# Security of Proof-of-Personhood: Idena

Jordi Subirà-Nieto

Supervisors: Prof. Bryan Ford, Louis-Henri Merino, Haoqian Zhang

*Decentralized Distributed Systems Laboratory, EPFL*

June 11, 2021

## 1 Abstract

Proof-of-Personhood (PoP) can be considered the ideal form of consensus algorithm to be used in permissionless distributed ledgers. Any system which unequivocally maps users to real human identities could implement a truly democratic and equally distributed governance. Recently, several approaches, aiming to achieve PoP, have been proposed.

In this semester project, we analyze the *Idena* protocol. More concretely, we focus on assessing the Sybil resistance of the Idena protocol, under some assumptions, against different attackers. We create several models which we use to simulate various attacks against the Idena network. Finally, we discuss some experiment outcomes that show how the Idena protocol falls short of accomplishing Sybil resistance for some of the proposed attackers.

## 2 Introduction

Proof-of-Personhood [1] (PoP) is an innovative concept that stems from the need to provide some alternative means of governance to permissionless Blockchains. In essence, PoP consensus algorithms aim to uniformly redistribute the network governance among all the users who compose it, rather than distributing it based on some proof of investment, as in Proof-of-Work (*e.g.*,the one used by Bitcoin [2]) or Proof-of-Stake (*e.g.*,utilized by Ethereum [3]). In order to achieve their goal, PoP-based systems injectively match one node in the network to one human identity, meaning that one human can only own one node.

Idena [4] is a permissionless Blockchain project based on PoP. Therefore, Idena guarantees that every node has the same chance to mine blocks, which implies that every node contributes equally to the network governance.

The Idena PoP protocol uses a Reverse Turing test (called FLIPS "Filter for Live Intelligent People") to grant a unique proof of humanity to its participants. In essence, these so-called flips are an extension of CAPTCHAS [5], which consists of two sequences of 4 pictures, one of them creating a meaningful story whereas the other one is just a shuffled sequence. This is believed to be an *AI-Hard* problem, although a doable task for humans.

In addition, the Idena protocol also tries to establish a certain Web-of-Trust in the system by issuing invitations every *epoch*, which is the time interval between validation ceremonies (*the Idena protocol is discussed in section 4*). New users can only join the network if they have received a valid invitation. Idena tries to motivate users to give invitations away to users that they know rather than sharing this invitation with anyone else. Inviters will receive an incremental amount of UBI (Universal Basic Income) for each one of the three subsequent rounds that the invitee pass. Proving the effectiveness of this economic model is hard; however, one might get the intuition that currently is pretty easy to gather several invitations by joining the Telegram channel [6] that Idena uses to interconnect users.

Ideally, any PoP network aims to be resistant to Sybil attacks [7], otherwise, some actor controlling a significant part of the network could compromise the system. Idena is not an exception and, thus, faces several challenges.

The first one is providing FLIPS which are *AI-hard*, at least, to adversaries who have access to cutting-edge AI technology. Willing to meet this hardness condition, the Idena protocol requires that every participant submit some flips periodically. The flips are based on some random words that provide context to create a meaningful narrative (*the random assignment is discussed more in-depth in the following section 4*).

The second one is being resistant to Sybil attackers, who adapt the number of resources used to grow over several rounds, obtaining some non-negative profit doing so. Among the resources that a Sybil attacker might use, there exist human workforces, e.g. Amazon Mechanical Turk [8]. This enables the attacker to hire a pool of workers to carry out some tasks. For the rest of this project, we will refer to those workers as *minions*. This concept was utilized by Ford [9] while evaluating the challenges that some PoP systems must deal with.

While the first one is an ongoing interesting research area, in this project, we will focus on assessing the security of Idena against Sybil attackers who use minions, while discussing the assumptions and the outcomes for different attacker models and network models.

In this project, we combine several attacker models with two different network models: a saturated network and a growing one. To carry out the experiments, we follow two distinct methods: a numerical approach based on some probability calculations and a simulator-based approach with two different levels of precision.

# 3 Background

Some interesting papers analyzing current approaches to achieve PoP consensus can be found in the literature. In *Identity and Personhood in Digital Democracy* [9], Ford analyzes a number of recently proposed approaches, while comparing them to Pseudonym parties. In the paper *Who Watches the Watchmen* [10], Siddarth et al review some existent solutions which try to achieve Sybil resistance by leveraging PoP.

# 4 Idena protocol overview

This section aims to summarize the Idena protocol and highlight the most relevant parts of it for this project. A more extensive description can be found in the Idena Whitepaper [11].

## 4.1 User status flow diagram

Users in the Idena network may obtain different statuses in every epoch, based on their past performance. At the end of the validation ceremony, the network reaches consensus about the status of every node. Figure 1 depicts the status flow for one user depending on her scoring and participation in the last rounds.

For the purpose of this project, it is interesting to distinguish between newbies and verified/human nodes. The former are temporal valid nodes, which means that they are not eligible to be granted invitations. In contrast, verified or human users will receive a certain number of invitations depending on their total scoring and the total issued invitations for a specific round, *i.e.*,the higher the scoring for a node the more invitations this node may receive.

Candidates are new users (or killed users) trying to join the network. In order to obtain candidate status, this user must have received an invitation from a verified or human user.

One node can be suspended or killed, either if the user does not participate in the validation ceremony for the next epoch or if the user does not solve enough flips to achieve a minimum

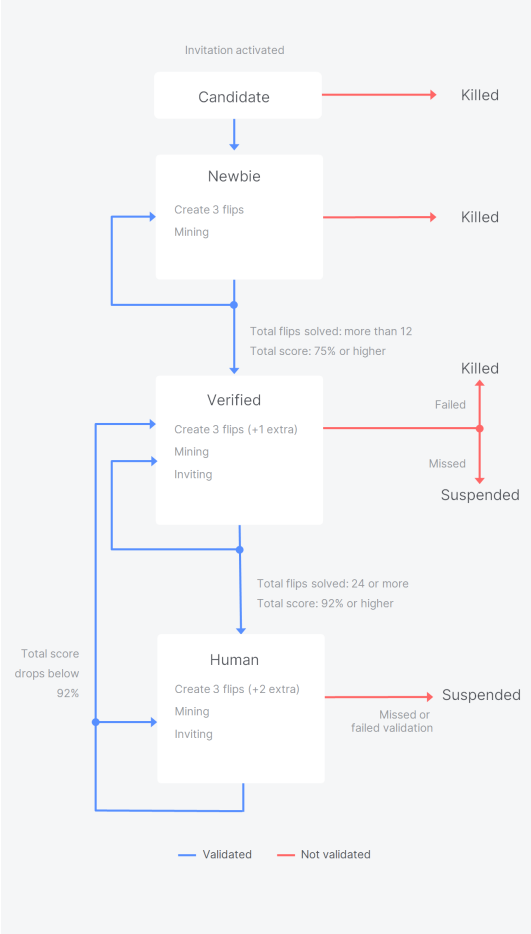



Figure 1: Idena status flow ( from [12])

threshold. It is also worth mentioning that suspended and killed nodes do not participate in the network for the current epoch, *i.e.*,neither they receive any UBI nor they can submit flips for the next validation phase. Similarly, candidates are nodes that have not been part of the network yet. Thus, for the rest of this project, we will refer to the number of alive nodes (*i.e.*,newbies, validated nodes and humans) as the network size.

## 4.2 FLIPS

As previously discussed, a FLIP is composed of two sequences of 4 pictures, one of them creating a meaningful story, whereas the other is just a random permutation of the former. Every flip is associated with a pair of words that provides context to create the narrative (*a further discussion about the word-pair assignment can be found in the following section 5*). Additionally, every user must have submitted 3 flips to the Idena Network to participate in the next validation ceremony. Otherwise, the protocol will consider as if the user had missed the validation ceremony.

The correct solution for a flip is reached by consensus, *i.e.*,considering all the answers provided by the nodes for that flip during the validation ceremony. Moreover, users solving a flip are entitled to report it, if either it contains an explicit ordering as part of the image (e.g. a sequence of numbers) or if the flip is not related to the word pair. In the case that enough users report a flip, the Idena protocol will discard the flip (*i.e.*,the flip will not be part of the grading) and the author will not receive any UBI for that flip.

## 4.3 Validation ceremony

The validation ceremony is a periodic and synchronous process (*i.e.*,all users will participate at the same time) that aims to renew the crypto-identity validity for every user. Users who successfully go through the validation ceremony will be validated to participate in the network until the next ceremony. This period is known as *Epoch*.

The validation ceremony is composed of several parts or phases:

### 4.3.1 Flip submission phase

This phase can be considered a setup phase rather than being part of the validation ceremony, strictly speaking. Every user willing to participate in the next ceremony must submit 3 flips to the Idena network. This phase is asynchronous, in contrast to the rest.

### 4.3.2 Lottery phase

The lottery phase randomly distributes flips among the users participating in the validation ceremony. In essence, every user will receive 6 flips to be solved during the short phase and a larger number of flips (about 20 of them) to be solved during the long phase. (*This distribution is discussed in more detail in the next section 5*).

### 4.3.3 Short phase

In the short phase, every node must solve 6 flips in 2 minutes. This short time restricts that one human can solve the short phase for more than one node. This restriction effectiveness is hard to prove since how fast a human can solve flips will depend on the cognitive capacity of that human to solve this task.

### 4.3.4 Long phase

During the long phase every node must qualify several flips (about 20 of them). The duration is considerably longer than in the short phase, 30 minutes. In order to qualify flips, the users must provide the correct answer for each flip and, on top of that, the user will be able to report flips.

After the long phase the network reaches consensus about the status for every flip, which can be:

- Strong consensus; if 75% of the users who solved that flip agree on the answer.
- Weak consensus; if 66% of the users who solved that flip agree on the answer.
- DQ; if less than 66% of the users who solved that flip agree on the answer.
- Reported; if more than 50% of the users who solved that flip report the flip in the long phase.

DQ and reported flips are not considered for the classification phase. For our purposes, strong and weak consensus will be equivalent, *i.e.*,the flip will be considered.

### 4.3.5 Classification phase

After all the users have completed the long phase, the network reaches consensus about the status for every node (after computing how many flips this user answered correctly). The status for every node might vary depending on the last validation ceremony scoring and the accumulated statistics for the previous ones, according to the status flow depicted in the Figure 1.

#### 4.3.6 Invitation distribution

Idena issues half of the network size invitations and distributes them to the non-temporal nodes (*i.e.*,verified nodes and human nodes). In a nutshell, the higher the scoring a node has achieved (over several ceremonies), the higher the probability that this node is going to receive one or several invitations.

# 5 Heuristic analysis

This section tries to analyze the reasoning behind some parts of the Idena protocol, which might be relevant in terms of security. It is worth mentioning that we carried out neither any formal verification nor any static or dynamic analysis. We extracted the following information from manually inspecting the code [13] and from the Idena Whitepaper [11].

## 5.1 Randomness source

Idena uses information in the Blockchain and Verifiable Random Functions (VRFs) [14] to compute seeds for various random processes in the protocol.

In the lottery phase, the randomness source is coming from the current block seed. In turn, this seed is the output of a *VRF* computed by the node proposing the block over yet another seed based on the previous block metadata.

In the flip word distribution, the randomness source is coming from the current epoch seed. When ending a ceremony, the network nodes reach consensus about the next epoch parameters (e.g. time to next epoch, valid nodes, nodes that deserve rewards, among many other details). The block seed is the outcome of a *VRF* over the header data of the first epoch block.

## 5.2 Word-Pair assignment

The dictionary contains (at the moment that this document is been written) 3300 words. Each word is mapped to some description (or synonyms). The dictionary can be found in the Idena public repository [13].

The Idena protocol assigns 9 different word pairs to each node. The threats for this part of the protocol are, mainly, attackers who can influence the assignment. For instance, some lazy user who achieves that Idena always assigns the same pair of words, so that she can reuse flips from previous rounds. We discuss at a high level how Idena tackles this challenge.

Every node computes 9 pairs of indexes randomly from 1 to 3300 (without repetitions). These 9 pairs are the final output of a sequence of VRF functions that starts with the Epoch seed as input. As aforementioned, the Epoch seed is shared by all the nodes and can be considered to have enough entropy. The VRF outputs a cryptographic proof as well. Upon the flip submission, the node will also submit the cryptographic proof and the intermediate seed used to generate the words. This cryptographic information allows the recipient network nodes to recover the word pairs and validate the transactions.

## 5.3 Flip submission and endorsement

Each validated user in the network must create, at least, 3 fips to participate in the next validation ceremony. The flip payload (i.e. the pictures) is encrypted with two keys. The first key encrypts the public part, which will be publicly available after the validation ceremony (2 images). The second key encrypts the hidden part, which will be available only to participants who solve the flip. This mechanism thwarts attackers who try to collect many flips to generate a database to train AI systems.

The nodes broadcast each flip to the Blockchain within a signed transaction. The network nodes endorse that this transaction is correct, *e.g.*,the flip is not duplicated or the word pair is honestly generated.

### 5.4 Lottery phase

Every node in the network computes the lottery flip assignment, *i.e.*,every node in the network knows to which users the flips from a specific author were assigned.

The lottery phase mapping is then a deterministic process, meaning that every node will compute the same mapping. This process uses a sequence of random permutations, while also considering some other factors to modify it slightly (*e.g.*,the candidate assigned to a flip cannot be the author for that flip).

During the short and long phase, the flip author sends the second key encrypting the flip only to those nodes that must solve that specific flip.

## 6 Network models

### 6.1 Network models overview

This project analyzes two types of networks: one saturated network and one growing network.

The former is saturated with respect to the legitimate nodes that compose the network, *i.e.*,the amount of legitimate nodes remains stable during subsequent rounds. In other words, the birth and death rate are considered equal for the non-Sybil nodes

The Idena network is intended to match one identity to one real person ( this also applies to other PoP networks), so it will stop growing when it gets near to the population size (whatever this population is). Studying this network model provides insight into how an attacker could harm this type of network.

On the other hand, the second network model grows, with respect to the legitimate nodes, at every round. This model captures the current state of Idena, *i.e.*,more users are joining the network, increasing the number of active nodes. To simplify our model we will assume a constant growth ratio.

Intuitively, it must be harder for an attacker who aims to take over the network to attack the growing network, since, in essence, this attacker must compensate for the fact that legitimate nodes also grow.

### 6.2 Network model 1

The first network model considers a stabilized network, i.e. the number of legitimate nodes that do not pass a validation phase is the same amount as the nodes that join or rejoin (if they had been suspended) the network. Thus, the network will only grow by the minions joining the network.

Although it is a saturated model, several invitations are issued at every round to satisfy a certain birth rate, being:

$I = Network\_size * Inv\_ratio$

### 6.3 Network model 2

The second network model considers a growing network, *i.e.*,the number of legitimate nodes will grow at every round by a specified ratio (we break down the growth ratio into node categories):

- Non-sybil humans growth = 0.0886
- Non-sybil newbies growth = 0.0616
- Non-sybil validated growth = 0.0846

We choose to fix the growth ratios to the previous values. Those values are a trimmed mean over the data extracted from Idena's statistics website [15].

The number of invitations issued at every round being:

$I = Network\_size * Inv\_ratio.$

Note that $Inv\_ratio$ can be bigger than the legitimate node growth (*e.g.*,some invitations might not be used).

### 6.4 Discussion about invitations

We will consider the invitations $I$ issued at every round to be:

$I = NW\_size * (inv\_ratio)$

Thus, the expected number of invitations that the Sybil attacker will receive (by collecting the distributed invitations to the Sybil nodes) in one round can be computed as:

$E[I] = Sybil\_node\_human\_status * (inv\_ratio)$

On top of the expected invitations, the experiments also consider that the attacker can persuade a certain amount of non-Sybil nodes (*i.e.*,legitimate nodes), so that those nodes give their invitations to the Sybil attacker. We will quantify that by defining a $Persuaded\_ratio$ parameter.

Then, the expected extra invitations that the Sybil attacker will collect from the rest of the network is defined as:

$E[Extra\_I] = Legitimate\_human\_status * (inv\_ratio) * (Persuaded\_ratio)$

## 7 Attacker models

*In this section, we will describe every Attacker model at a high level, so that the reader can get the gist of them without being confused with too many details. A more low-level description, the algorithm pseudo-code and the probability calculations for every attacker can be found in the Appendix section A.*

In this project, we will consider 4 incremental attackers regarding their strategies, *i.e.*,attacker 2 will follow a strategy that will allow her to take over the network faster than attacker 1.

The attackers share the same aim, which is taking over the network. For this project, taking over the network is equivalent to control 33% of the network nodes. As stated in the Idena website [12], the network would not be able to recover after an attacker has been able to control one-third of it.

### 7.1 Assumptions and capabilities

The attackers also share the same capabilities in terms of both: investment capacity and computational resources.

The main tool for the attackers will be *minions*. *Minions* are human resources hired, on-demand, to solve the validation ceremony on behalf of the attacker.

All the attackers are able to invest in some initial *minions* which will be able to validate $N_0$ (a parameter for the different experiments) initial Sybil nodes. In subsequent rounds, the attackers will use the UBI (which includes ceremony rewards and mining rewards) earned in the previous round to hire $N_i$ *minions* for the current round.

Similarly, the attackers possess an automatic process (bot) with access to a common Sybil flip database (containing all Sybil flips for a given round). This bot does not have any AI capability which allows it to distinguish non-Sybil flips better than randomly. However, this bot is able to solve successfully any Sybil-flip, since it has access to the database. In other words, when this bot is assigned one non-Sybil flip, it will choose between the left or the right option at random and when assigned a Sybil flip, it will choose the correct answer for that flip.

For most of the models, we have to assume that the attacker has enough time and/or computational resources to check the flip assignment to every Sybil node and to distribute the minions among the nodes accordingly to the algorithm used by the specific attacker model. The following example will hopefully clarify this assumption.

Imagine that the attacker $A_i$ controls 8 nodes that participate in the current validation ceremony and the attacker can use 5 minions. The previous assumption implies that $A_i$ can decide, in a negligible amount of time (compared to the short validation phase duration), which nodes are assigned to those 5 minions, depending on the algorithm used by $A_i$.

For the sake of simplicity, we also assume that all flips include the correct answer, *i.e.*,the one that the author of that flip intends to create as the correct one. This differs from the Idena protocol, in which the correct answer for a given flip is based on the consensus reached in the long phase of the validation ceremony. This simplification is acceptable since we assume that the flips are correctly generated (*i.e.*,any user can distinguish clearly the meaningful and the meaningless story) and the human users choose the correct answer for any flip.

## 7.2 Minion hiring model

Minions are a fundamental tool for the attacker which enables him to attack the network and, in some cases, succeed while doing so. However, minions also incur an associated cost. For the models described in this project, the attacker uses the UBI earned in one round to hire minions in the next round. In addition, we assume that the minion-hiring friction is 0, *i.e.*,the minion is paid the same amount of UBI as the Sybil attacker receives from the network for one node.

One of the questions which might arise at this point is whether there exists a real incentive for the minions to be part of the Sybil network. This is not a trivial question, since it depends on the economic and social assumptions made. For instance, it might be more convenient for a user to be a minion since she will receive the same UBI but she does not have to care about hosting (and even paying for the equipment) and maintaining the node up. Similarly, minions might not need to bother to create flips since this could be done separately by the attacker. Further discussion and analysis are considered out of the scope of this project and left as future work.

The reader might also wonder whether there is a more profitable way to take over the network, for example, by using minions only to solve ceremonies for a fraction of the Sybil nodes, *i.e.*,employing fewer minions throughout the attack. We will use the Figure 2 to discuss this question.

The plot depicts the numerical solution for the next equation (the invitation ratio is fixed at 0.5):

$$Pr[Sybil\_ratio] * (E[I] + N * (1 - \alpha)) + N * \alpha >= N;$$

$$where : N := sybil\_nodes; E[I] := Inv\_ratio * N; \alpha := minion\_ratio;$$

This equation captures the relationship between the minion ratio (the percentage of minions used for the Sybil nodes) and the Sybil ratio (the Sybil nodes compared to the network size), such that in the next round, the attacker will control, at least, the same number of nodes.

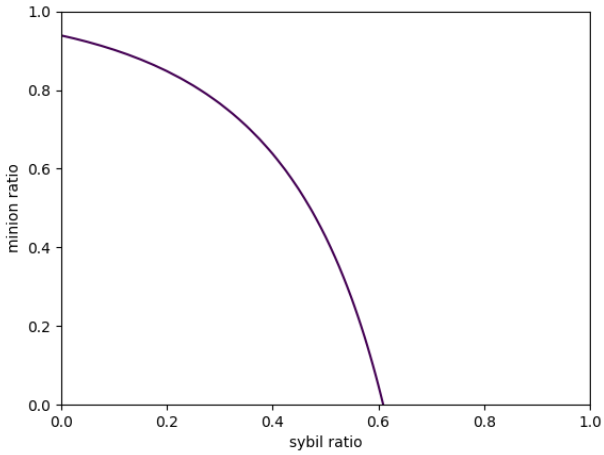



Figure 2: Minions and Sybil nodes relationship

The solution for the previous inequality lies in the right-sided area of the plot. It can be noticed that if the attacker controls a moderate ratio of the network, *e.g.*,10%, she needs to hire minions for more than 90% of her nodes. In addition to this margin being small, by using fewer minions the attacker will also grow more slowly, meaning that it will take more rounds for her to succeed.

Intuitively, it seems more interesting for the attacker, to grow as fast as she can until reaching her goal (*e.g.*,one-third of the network) and, afterward, remaining stable and accumulating, indefinitely, profit from the network by using the strictly necessary percentage of minions (*i.e.*,$\alpha$).

## 7.3 Consideration about the experiments

In the following sections, *rounds* represents validation ceremonies in the Idena protocol. Thus, in terms of time, one round will be equal to the period between two ceremonies, which is roughly 1 month.

The mechanism that was used to produce the experiment outcomes (*i.e.*,the numerical approach, the full simulator or the lightweight simulator) is explicitly stated for each one of them. We will use the probability models to plot the first three attackers while using the simulators for the last one. This is due to the fact that the first three attackers can be reasonably approximated using some probability and expectation computations, while this would be more contrived for Attacker 4. In any case, similar results can be obtained for the first three attackers by running the simulations. *The reader can also find some experiment examples that serve as a comparison between the simulators and the probability computations. These results show that similar outcomes can be achieved by using any of the tools.*

### 7.3.1 Wining condition for the attacker

We will consider that one attacker is successful if she can reach her goal, *i.e.*,controlling one-third of the network nodes, in a realistic amount of time. Previously, we already assumed that the initial investment is affordable to all the attackers.

For this project, we do not require any minimum percentage of the Sybil identities to be non-human at the end. We rather analyze how the attacker can leverage the revenue system to take over the network. Nonetheless, as briefly discussed before when we analyzed Figure 2,

this percentage can be computed and it will depend on the network parameters. However, we abstract the experiments from this complexity.

## 7.4 Attacker 1

*For a more schematic explanation, the pseudo-code for the algorithm used by this attacker and also the probability calculations, please refer to the following section in the appendix A.1.2.*

As for every other attacker, we assume that Attacker 1 controls $N_i$ nodes at the beginning of the validation epoch $i$. This means that the attacker can hire also $N_i$ *minions* for the current epoch. We also assume that the attacker can activate $I_i$ invitations, *i.e.*,$I_i$ Sybil nodes with candidate status.

This attacker hopes for a *lucky assignment* happening for any of the $N_i + I_i$ nodes. We define this lucky assignment as one node receiving 5 or 6 Sybil flips (out of the 6 flips assigned in the short validation phase). The attacker will use the bot process mentioned before to deterministically solve the validation ceremony for that *lucky* node.

The attacker's strategy is pretty simple and consists of using as many *minions* as needed to validate the old nodes ($N_i$ nodes for this round) and using the rest to validate some of the $I_i$ candidates. Thus, at the end of the $i-th$ round the attacker will control $N_{i+1}$ nodes, where:

$N_{i+1} = N_i + lucky\_nodes.$

**Experiment 1 (Probability computation)**

Expectation over the number of rounds to take over the network using Attacker 1 depending on the initial amount of Sybil minions $N_0$ used by the attacker considering different invitation ratios.

- Uses Network Model 1
- Inital network size = 1000 + Initial minions

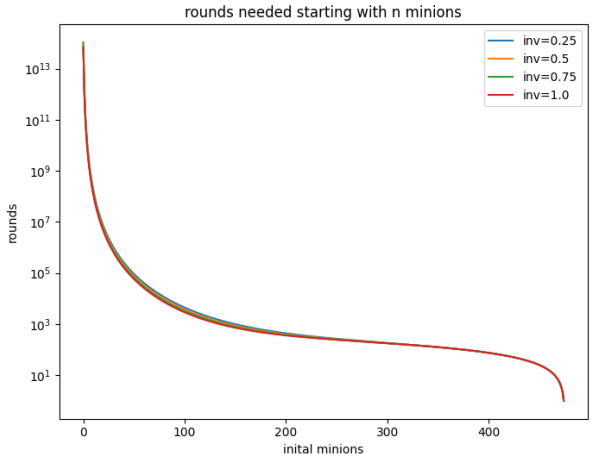



Figure 3: Attacker 1, Experiment 1

We can extract some interesting information from this graph:

The first aspect to mention is that the time to take over the network exponentially decreases with respect to the percentage of the network that the attacker controls. Note that a log scale is used in the plot.

The second aspect to notice is that for the Attacker 1 there is not a substantial difference in the number of rounds to take over the network between different invitation ratios.

The third interesting piece of information that we can extract is that the attacker will eventually take over the network for any amount of initial minions, although, in practice even controlling, for instance, a tenth of the network size (*i.e.*,100 nodes) the attacker will need thousands of round to take over. We can roughly relate one epoch to one month, which would mean that the attacker would need several decades (even centuries) to achieve her goal.

**Experiment 2 (Probability computation)**

Expectation over the number of rounds to take over the network using Attacker 1 depending on the initial amount of Sybil minions $N_0$ used by the attacker considering different persuasion ratios (hoarding ratio in the plot).

- Uses Network Model 1
- Inital network size = 1000 + initial minions
- Invitation ratio = 1.0

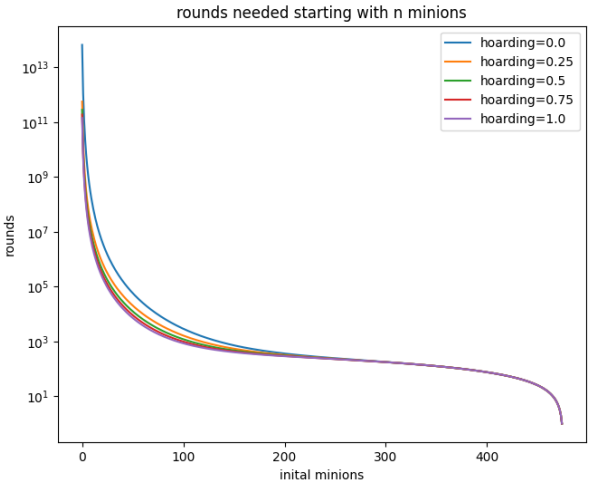



Figure 4: Attacker 1, Experiment 2

Considering that the adversary can persuade some legitimate (non-Sybil) nodes so that they give away invitations to the Sybil attacker decreases the number of rounds needed to take over the network. Even though, in practice, it is still hard for the attacker to achieve her goal in a reasonable amount of time if the fraction of nodes that she controls is not close to the target ratio.

**Experiment 3 (Probability computation)**

Number of rounds to take over the network using Attacker 1 for the following experiment:

- Uses Network Model 2
- Inital network size = 333
- Initial minions = 100
- Invitation ratio = 1.0
- Persuaded ratio = 1.0

For the growing network model, it becomes quite clear that even assuming a quite optimistic scenario for the Attacker 1 (*i.e.*,the initial amount of nodes near to the target ratio, 33%, high

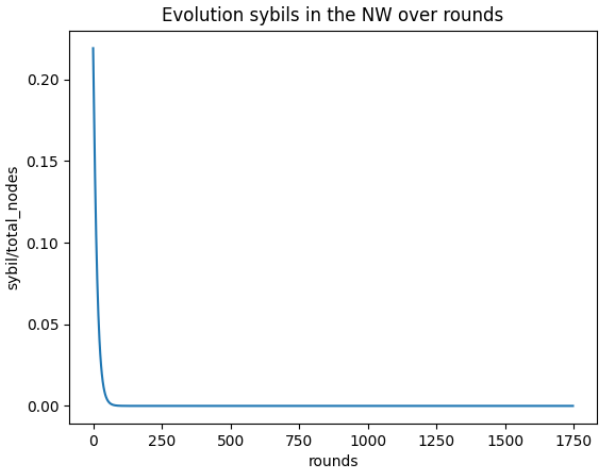



Figure 5: Attacker 1, Experiment 3

invitation and persuaded ratio), the ratio of controlled nodes, with respect to the network size, rapidly plummets. In other words, the attacker will never succeed to take over the network.

## 7.5 Attacker 2

*For a more schematic explanation, the pseudo-code for the algorithm used by this attacker and also the probability calculations, please refer to the following section in the appendix A.1.3.*

Attacker 2 uses a different approach which leverages the fact that bots can be used to non-deterministically solve validation ceremonies for the candidate nodes.

Thus, this attacker's strategy consists of using bots to solve the validation ceremony for the $I_i$ candidates. As aforementioned, the bot will choose between the left or the right option for a flip with the same probability, meaning it will choose the meaningful story for half of the legitimate flips. Therefore, depending on the Sybil flips assigned every bot will be able to solve the validation ceremony with a certain probability ($Pr[Z]$ in the probability discussion subsection for this attacker A.1.3).

At the end of the round $i$ the attacker will control $N_{i+1}$ nodes, where:
$N_{i+1} = N_i + successful\_bots$

**Experiment 1 (Probability computation)**

Expectation on the number of rounds to take over the network using Attacker 1 depending on the initial amount of Sybil minions $N_0$ used by the attacker considering a fix invitation ratio.

- Uses Network Model 1
- Initial network size = 1000 + initial minions
- Invitation ratio = 0.1

This plot shows an exponential decay in the number of rounds that the attacker will need to take over the network with respect to the percentage of controlled nodes in the network. Even considering that Attacker 2 controls only 3 nodes (0.3% of the initial network size), she will be able to take over the network in a few decades (about 30 years). This amount of time might be considered non-practical in a real attack-case scenario.

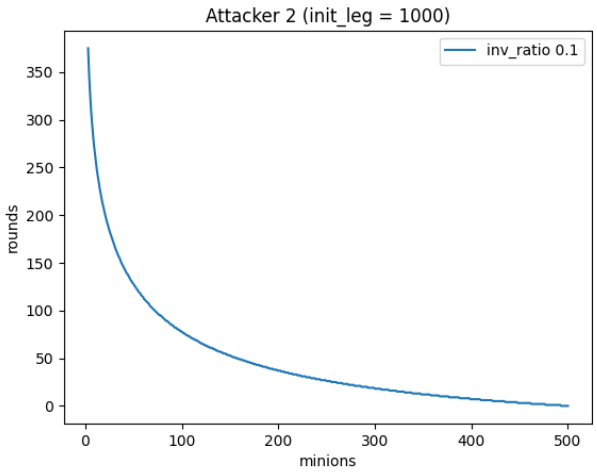



Figure 6: Attacker 2, Experiment 1

**Experiment 2 (Probability computation)**

- Uses Network Model 1
- Initial network size = 1000 + initial minions

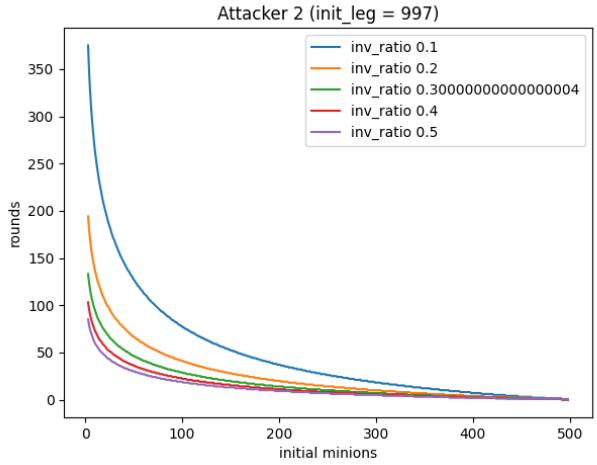



Figure 7: Attacker 2, Experiment 2

This graph shows that the number of rounds significantly decreases as a larger number of invitations are issued at every round. This makes sense, since the expected number of invitations that the Sybil nodes will receive will be larger, and so will the expected number of successful bots.

In contrast to the previous plot, in this one, we can already see that for some invitations ratios the attacker can take over the network in a pretty reasonable amount of time (a few years).

**Experiment 3 (Probability computation)**

- Uses Network Model 1
- Initial network size = 997 + initial minions
- Invitation ratio = 0.1

Yet another interesting experiment which illustrates that Attacker 2 can effectively improve her performance (in terms of reducing the time to take over the network) by persuading only a

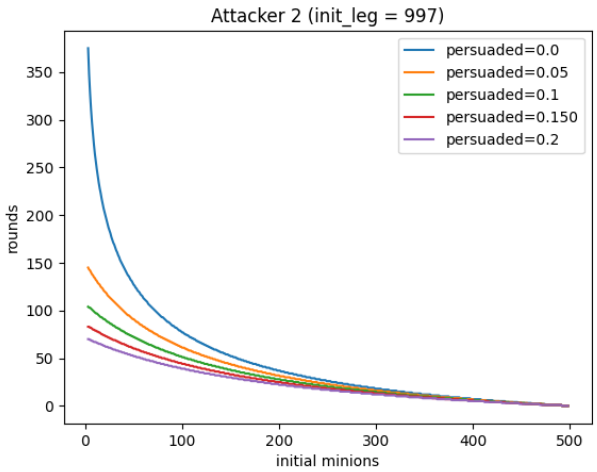



Figure 8: Attacker 2, Experiment 3

tiny fraction of non-Sybil nodes so that they share their invitations with the Sybil attacker. For instance, by convincing a 10% of the non-Sybil users who received an invitation (*i.e.*,gathering an extra 1% of the invitations, since the invitation ratio is 10%) the attacker reduces the number of rounds needed from more than 350 down to roughly 100 hundred rounds.

**Experiment 4 (Probability computation)**

- Uses Network Model 2
- Initial network size = 333
- Initial minions = 33
- Invitation ratio = 0.5

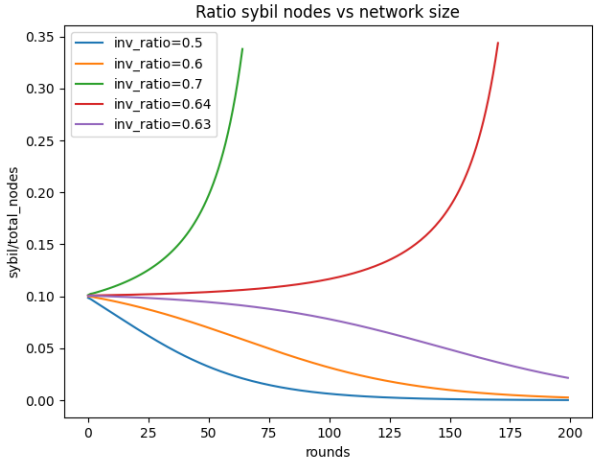



Figure 9: Attacker 2, Experiment 4

This plot shows the Attacker 2, initially controlling 10% of the network, trying to take over a growing network (Network Model 2) for different invitation ratios. The main takeaway is that for a certain invitation ratio value, more concretely below 0.64, the Sybil-nodes to Total-nodes ratio decreases over rounds, meaning that the attacker can never succeed.

**Experiment 5 (Probability computation)**

- Uses Network Model 2

- Initial network size = 300 + Initial minions
- Invitation ratio = 0.5

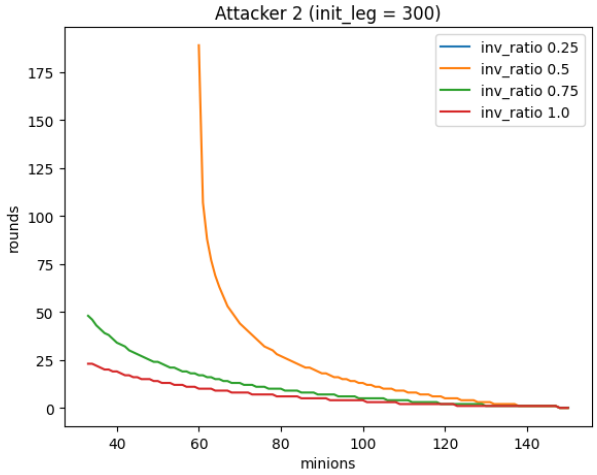



Figure 10: Attacker 2, Experiment 5

We now take a look at the general case. Firstly, it is worth mentioning that the blue line for the 0.25 invitation ratio case is not plotted since the attacker will never accomplish her goal for this parameter. Secondly, if we take the 0.5 invitation ratio case, we can see that the curve follows an asymptote near to the 60 minions value. The interpretation is that the attacker will never be successful (independently of the number of rounds) for any initial minions value below 60.

**Experiment 6 (Probability computation)**

- Uses Network Model 2
- Initial network size = 300 + Initial minions
- Invitation ratio = 0.5

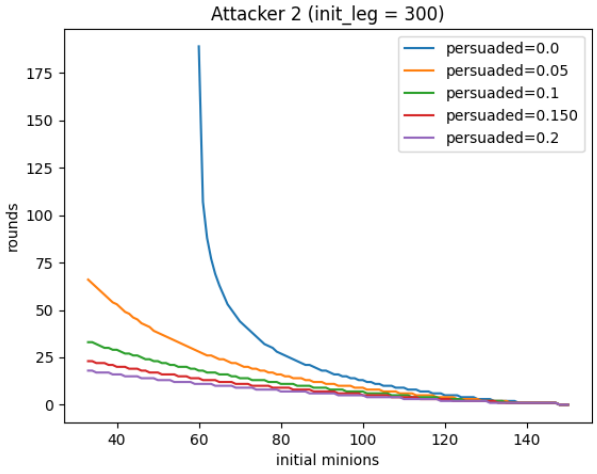



Figure 11: Attacker 2, Experiment 6

In this plot, we fix the invitation ratio to 0.5 and we plot the blue line as a baseline. We can see that with a moderate persuasion capacity, the attacker can significantly improve her performance, being able to take over the network in a reasonable number of rounds while only controlling a small part of the initial network.

## 7.6 Attacker 3

*For a more schematic explanation, the pseudo-code for the algorithm used by this attacker and also the probability calculations, please refer to the following section in the appendix A.1.4.*

Attacker 3 slightly improves Attacker 2 by merging the strategy from Attacker 1, *i.e.*,this attacker will use bots to solve the validation ceremony for the old nodes ($N_i$ many for the current round $i$) which have been assigned 5 or 6 flips. This means that *minions* can be used to deterministically validate some candidates, in addition to using bots for the rest of them.

At the end of the round $i$ the attacker will control $N_{i+1}$ nodes, where:
$N_{i+1} = N_i + lucky\_old\_nodes + successful\_bots$

As aforementioned, Attacker 3 slightly improves Attacker 2, and the general behavior that was analyzed for the latter also applies to the former. For the sake of completeness, we will briefly comment on a subset of the experiments discussed for the previous attacker. *Please, refer to the end of this section 7.8, in order to compare outcomes for the same experiment and different attackers.*

**Experiment 1 (Probability computation)**

- Uses Network Model 1
- Initial network size = 997 + initial minions
- Invitation ratio = 0.1

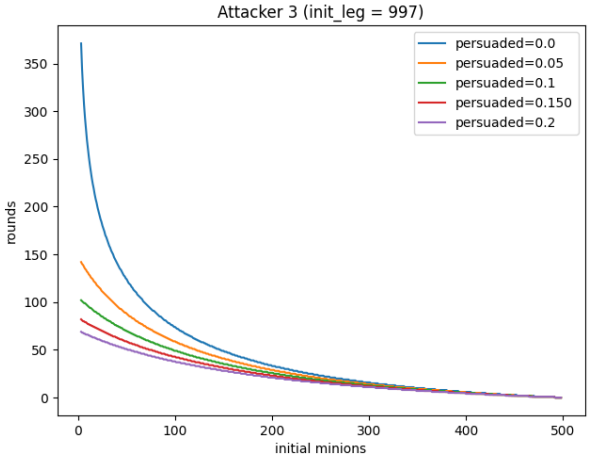



Figure 12: Attacker 3, Experiment 1

**Experiment 2 (Probabilistic approach)**

- Uses Network Model 2
- Initial network size = 300 + Initial minions
- Invitation ratio = 0.1

As expected, the trend for both experiments is very similar to the same experiments conducted for Attacker 2 and only the absolute values slightly differ from each other. Therefore, we can conclude that Attacker 3 achieves a minor improvement compared to Attacker 2.

## 7.7 Attacker 4

*For a more schematic explanation, the pseudo-code for the algorithm used by this attacker and also the probability calculations, please refer to the following section in the appendix A.1.5.*

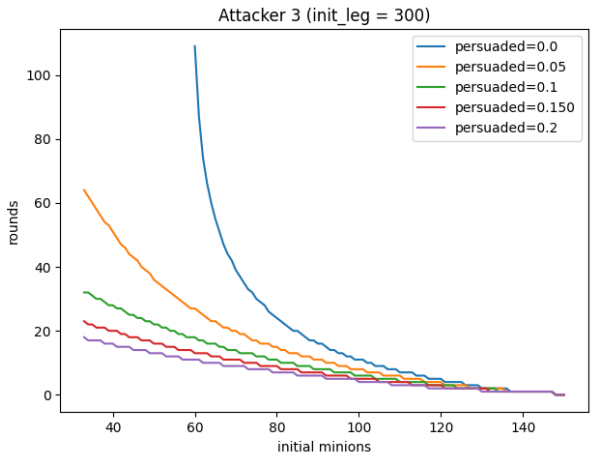



Figure 13: Attacker 3, Experiment 2

Attacker 4 goes one step further in her strategy and assigns the available minions (which are equal to the amount of Sybil nodes at the beginning of the round, $N_i$) to her nodes prioritizing the ones that have been assigned fewer Sybil flips. Intuitively, one can see that those nodes which have been assigned a fewer number of Sybil flips are less likely to be validated by the bot process.

In other words, the attacker will sort the nodes increasingly (by the amount of Sybil flips assigned) and will distribute minions accordingly. Once the attacker runs out of minions, the bot process will carry out the validation ceremony for the remaining nodes. assign

At the end of the round $i$ the attacker will control $N_{i+1}$ nodes, where:
$N_{i+1} = N_i + successful\_bots$

**Experiment 1 (LW simulator)**

- Uses Network Model 1
- Initial network size = 1000 + Initial minions
- Invitation ratio = 0.1

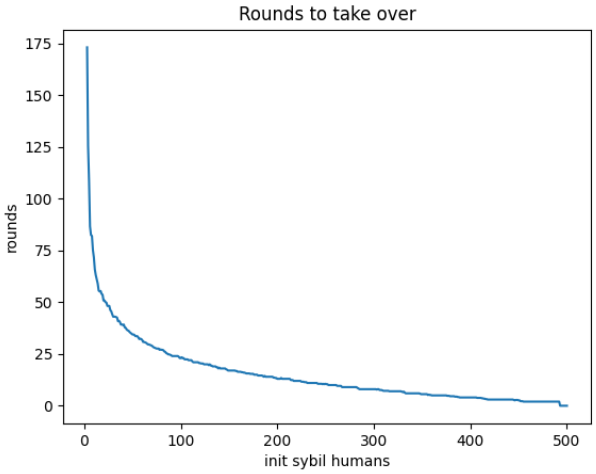



Figure 14: Attacker 4, Experiment 1

For this experiment (stable network), we can see that the attacker is successful for a quite moderate number of initial minions. For instance, for a few of them (10 minions, which account

for 1% of the initial network size), the attacker can take over the network in about 60 to 70 rounds.

**Experiment 2 (LW simulator)**

- Uses Network Model 1
- Initial network size = 1000 + Initial minions

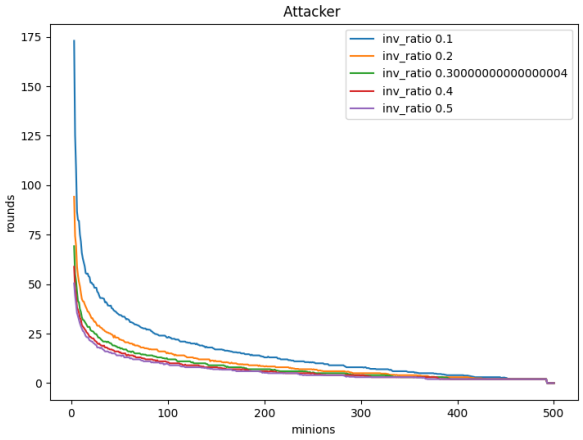



Figure 15: Attacker 4, Experiment 2

Similar to previous attackers, increasing the invitation ratio decreases significantly the number of necessary initial minions.

**Experiment 3 (LW simulator)**

We consider the growing network model for the following experiment

- Uses Network Model 2
- Initial network size = 300 + Initial minions

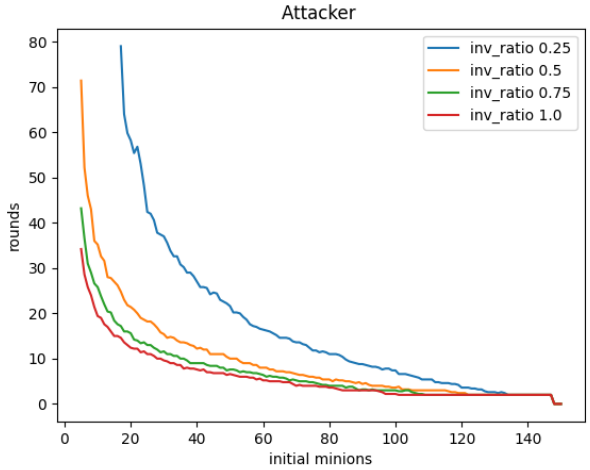



Figure 16: Attacker 4, Experiment 3

From this plot, it can be extracted that Attacker 4 is mostly successful even considering different invitation ratios. This provides an intuition about the improvement that this attacker

achieves. For instance, considering a 0.5 invitation ratio (which corresponds to the current value in the Idena protocol), an attacker who controls a small fraction of the initial network can take over a growing network, which also grows at a similar pace to Idena, in a few rounds.

**Experiment 4 (LW simulator)**

- Uses Network Model 2
- Initial network size = 300 + Initial minions
- Invitation ratio = 0.5

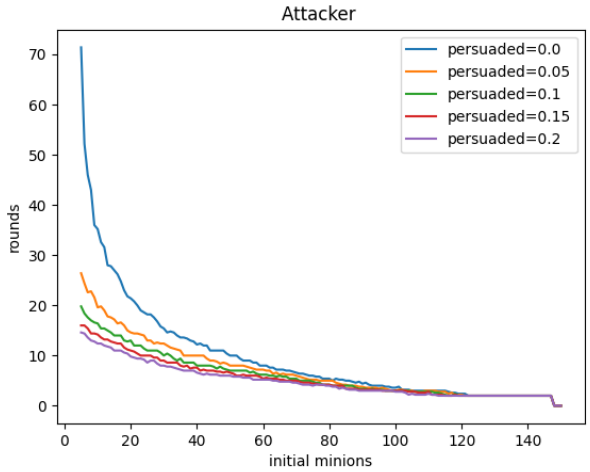



Figure 17: Attacker 4, Experiment 4

Similar to previous attackers, Attacker 4 can drastically reduce the number of rounds needed to take over the network when she persuades a tiny fraction of non-Sybil users.

## 7.8 Attacker comparisons

**Experiment 1 full simulator**

- Uses Network Model 1
- Initial network size = 1000
- Initial minions $N_0 = 100$
- Invitation ratio = 1.0

By comparing the four attackers, the reader can observe that the gap in the number of rounds to take over between Attacker 1 and the rest is vast. Thus, we will refrain to consider Attacker 1 for the rest of the comparisons.

**Experiment 2 full simulator**

The parameters defined for the Experiment 2 are:

- Network Model 1
- Initial network size = 1000
- Initial minions $N_0 = 3$
- Invitation ratio = 0.1

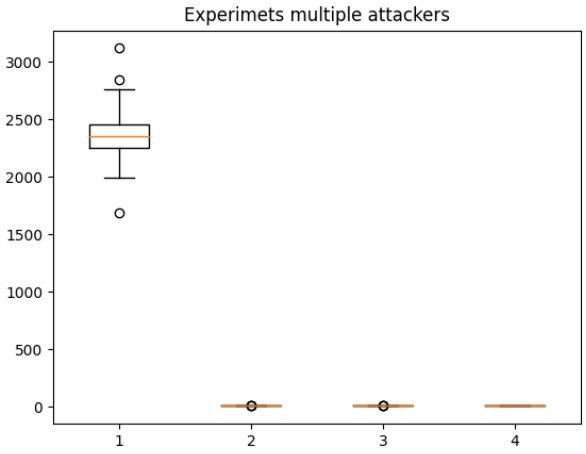



Figure 18: Comparison, Experiment 1.
Rounds to reach 1/3 of the NW

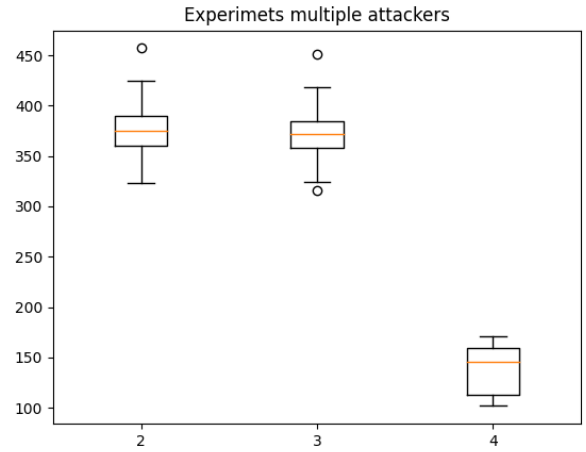



Figure 19: Comparison, Experiment 2.
Rounds to reach 1/3 of the NW

The main takeaway from this plot is that Attacker 3 slightly improves Attacker 2, while Attacker 4 cuts down the number of rounds needed to take over the steady network by more than half. In addition, some other information might be extracted as well.

Attacker 3 is the one with the lowest variance. This is due to the fact that this attacker deterministically solves for every old node (by using minions or bots which will always succeed) and with a certain probability, some candidates (new nodes entering the network) can be solved using minions (if some minion was not needed to solve the validation phase for old nodes).

In contrast, Attacker 4 is the one with the highest variance. The reason is that Attacker 4 might solve some validation ceremonies for old nodes non-deterministically, *i.e.*,using bots that might fail for this validation ceremony. If an old node is not validated, this node will not opt to receive an invitation from the Idena network.

### 7.8.1 Experiment 3 (Probability computation and LW simulator)

This table compares the minimum invitation ratio that every attacker would need to take over a saturated network in less than 200 rounds (considering that no legitimate node is persuaded to give invitations away).

- Uses Network Model 1
- Initial network size = 1000
- Initial minions $N_0 = 1$
- Maximum number of rounds = 200

| Attacker | min inv_ratio |
|---|---|
| 1 | - |
| 2 | 0.25 |
| 3 | 0.25 |
| 4 | 0.17 |

Table 1: Comparison, Experiment 3

As expected, Attacker 2 and Attacker 3 need the same invitation ratio, while Attacker 4 requires a smaller invitation ratio. Attacker 1 will never succeed (even if the invitation ratio were maximum).

#### 7.8.2 Experiment 4 (Probability computation and LW simulator)

Similarly, this experiment compares the minimum persuaded ratio (*i.e.*,the percentage of non-Sybil users who give invitations to the Sybil attacker at every round) that every attacker would need to take over the growing network in less than 100 rounds.

- Uses Network Model 2
- Initial network size = 1000
- Initial minions $N_0 = 1$
- Invitation ratio = 0.5
- Maximum number of rounds = 100

| Attacker | min persuaded_ratio |
|---|---|
| 1 | - |
| 2 | 0.11 |
| 3 | 0.09 |
| 4 | 0.01 |

Table 2: Comparison, Experiment 4

The reader can see that Attacker 4 needs to convince only 1% of non-Sybil nodes in order to achieve her goal. This gives an intuition about the Attacker's 4 effectiveness with respect to the rest.

## 8 Implementation

The source code for this project can be found in `https://github.com/dedis/student_21_pop_security/tree/main/idena_simulator`. It consists of various python scripts and modules which use Python 3.8.6 version and the third-party packages in the *requirements.txt* file.

The relevant files are:

*prob_models.py*, the script which carries out the probability computations and outputs the corespondent plots for the Attacker 1, 2 and 3.

*simulation.py* which contains the full simulator code.

*lw_simulation.py* which contains the lightweight simulator.

For this project, we created two simulators. The full simulator mimics the most relevant parts of the Idena protocol for this project, *i.e.*,the lottery phase and the short phase. The lightweight simulator approximates the lottery phase assignment by ignoring some lower-level details, such as considering less used authors for the current ceremony or distinguishing between flips for the same author. The full simulator is substantially slower than the lightweight simulator, especially for the growing network experiments. In any case, the growth for the growing network model is inevitably exponential (based on the assumption that the legitimates nodes grow by a constant multiplicative factor), so one cannot expect to conduct indefinitely long experiments in terms of rounds.

# 9 Future work

As mentioned in the Minion hiring model section 7.2, for this project we only consider a 0 friction model. However, it might be interesting to analyze the previous attacker models under different economic models and assumptions. The attacker models might serve as a basis to implement new models. For instance, it should not be too complicated to extend the models so that they can compute the total expenditure for some attackers, considering that the attacker pays some additional delta (with respect to the UBI) to every minion.

This project analyzed the security of Idena and the attackers were adapted to the Idena protocol, however, the ideas used for the attackers' protocols can be applied to any PoP network which leverages some form of threshold verification, such as BrightID[16] or Kleros[17].

In addition, new smarter attackers which increase their effectiveness can be suggested. Those new attackers could even use some AI-enabled techniques that enable bots to choose the answer for non-Sybil flips better than randomly. Similarly, new network models capturing more complex birth-death distributions could be analyzed.

# 10 Conclusion

The Idena network is a truly interesting open-source permissionless Blockchain which, at the moment, has thousands of active users. The project is actively maintained by several contributors who, regularly, release updates and patches. By using FLIPS and online ceremonies, the Idena protocol provides some flexibility compared to other approaches, such as Pseudonym Parties, while also providing user privacy, in contrast to other systems that require biometric information from the user. However, as seen in the previous sections, the Idena protocol is prone to Sybil attacks, due to several reasons.

On the one hand, the Idena validation ceremony still offers an opportunity for the attacker to be validated even if she uses dummy bot processes that randomly solve flips. As it was discussed, the risk is even bigger if this attacker coordinates her Sybil nodes and hires an adaptative workforce (also known as *minions*). In order to compensate for that, it might seem tempting to decrease the probability by requiring users to solve more flips. The problem lies in the fact that this will make the solving task more difficult for humans, increasing the false positive rate, meaning that more humans would fail on proving their human condition.

On the other hand, Idena relies on a Web-Trust model which assumes that users will only provide invitations to other users who they trust that will behave honestly. It is hard to prove that the incentive economic model of Idena accomplishes that. On the contrary, it is considerably easy to receive an invitation from some Idena users by joining the Idena Telegram channel [6].

To summarize, we have proved that, under some feasible assumptions, Idena does not guarantee resistance against some Sybil attackers.

# A Appendix

## A.1 Attacker models

### A.1.1 How to read the attacker models

The following sections follow the same pattern, meaning that they contain several redundant parts that are repeated between attackers. This is intended so that the reader can access the information for every attacker independently. However, the most relevant changes between attackers are highlighted, so that the reader can easily identify them.

### A.1.2 Attacker 1

**Aim**

Taking over the network (in this case controlling 33% of the network) by reinvesting the benefit from the network.

**Capabilities**

- The attacker has enough investment capacity to pay for the first nodes $N_0$ entering the network.
- The attacker has an automatic process (bot) with access to a common Sybil flip database (containing Sybil flips for a given round).
- The protocol execution time is negligible compared to the short ceremony duration. In practice, the bots and minions have enough time to solve the short validation phase after running the protocol.

**Protocol**

**Initial rounds:**

- The attacker initially hires $N_0$ minions (by making an initial investment).
- The minions will be assigned to nodes that will join the network and (after some initial rounds) that will be verified by the network (obtaining Human status).
- In every round, the minions are paid the UBI for that node (including ceremony rewards and mining rewards).
- After those initial rounds, the amount of available minions is $N_0$ ($M = N_0$), i.e. we can pay as many minions as validated Sybil nodes.

**Subsequent rounds:**

Setup phase:

- The available amount of minions $M$.
- The nodes controlled by the attacker $Sybil\_nodes = M$
- For each minion the attacker has to submit $f$ flips. Those flips are shared with the bot DB.
- The attacker gets $I$ invitations for a given round. The attacker activates those invitations.
- The amount of Sybil candidates in this round is $Sybil\_Cand = Sybil\_nodes + I$; where $Sybil\_nodes$ = those nodes validated in previous epochs (age >= 1) and I are new candidates (age = 0). We assign an index to each i.e. $i = 1, .., C$

Validation phase:

```
Validated\_nodes = 0
Available\_minions = M
For every c_i the attacker checks:
    If $c_i.flips$ contain 5 or 6 Sybil flips:
        The bot can safely solve the validation ceremony for that node
        This node is NOT assigned one minion
        Validated\_nodes++;
    Else if Available\_minions > 0:
        This node is assigned one minion, then
        Available\_minions--;
        Validated\_nodes++;
```

At the end of the validation phase:

- Sybil_nodes = Validated_nodes
- The attacker invests the network revenue in that epoch to hire minions for the next round, thus $M = Sybil\_nodes$

**Probability discussion**

We compute the probability of nodes getting Sybil flips assigned following a Binomial distribution.

- Short flips to be solved = 6
- X being the random variable which indicates how many Sybil flips are assigned to one node (out of 6):
- $Pr(X = k) = \binom{6}{k} * Sybil\_ratio^k * (1 - Sybil\_ratio)^{6-k}$
- $Pr(X >= 5) = Pr(X = 5) + Pr(X = 6)$.
- E[new sybil nodes] = $M * P(X >= 5)$ (M = sybil nodes at the beggining of the round).
- E[sybil nodes at the end of round i] = M + E[new sybil nodes]
- Following an iterative approach we can compute E[sybil nodes for round i+1]

### A.1.3 Attacker 2

This attackers shares aim and capabilities with the previous attacker but uses a different algorithm to increase to probability of succeed in every round.

**Aim**

Taking over the network (in this case controlling 33% of the network) by reinvesting the benefit from the network.

**Capabilities**

- The attacker has enough investment capacity to pay for the first nodes $N_0$ entering the network.
- The attacker has an automatic process (bot) that can access a common Sybil flip database (containing Sybil flips for a given round).

**Protocol**

**Initial rounds:**

- The attacker initially hires $N_0$ minions (by making an initial investment).

- The minions will be assigned to nodes that will join the network and (after some initial rounds) that will be verified by the network (obtaining Human status).
- In every round, the minions are paid the UBI for that node (including ceremony rewards and mining rewards).
- After those initial rounds, the amount of available minions is $N_0$ ($M = N_0$), i.e. we can pay as many minions as validated Sybil nodes.

**Subsequent rounds:**

Setup phase:

- The available amount of minions $M$.
- The nodes control by the attacker $Sybil\_nodes = M$
- For each minion the attacker has to submit $f$ flips. Those flips are shared with the bot DB.
- The attacker gets $I$ invitations. The attacker activates those invitations
- The attacker uses minions to solve flips for Sybil_nodes (old nodes) and uses bots for $I$ nodes (which will be nodes with candidate status, i.e. trying to join the network for the first time) .

Validation phase:

```
Validated_nodes = 0
Available_minions = M
For every c_i in Sybil_nodes:
        This node is assigned one minion, then
        Available_minions--;
        Validated_nodes++

# At this point: Validated_nodes = Sybil_nodes

For every c_i in I:
    If simulate_bot(c_i) == SUCCESS
        Validated_node++

def simulate_bot(c_i):
    solved_flips = 0
    For c_i.flips:
        If flip is a Sybil flip:
            solved_flips++
        Else
            if toss_coin() == 'HEADS':
                solved_flips++
    If solved_flips >= 5
        return SUCCESS
    Else
        return FAIL
```

At the end of the validation phase:

- Sybil_nodes = Validated_nodes
- The attacker invests the network revenue in that epoch to hire minions for the next round, thus $M = Sybil\_nodes$

### Probability discussion

We discuss the probability of one bot to pass the validation phase:

- We define a random variable Y_n being the legitimate flips correctly answered by one Bot node (out of n flips), following a Binomial distribution with p = 0.5:
  $Pr[Y_n = l] = \binom{n}{l} * 0.5^l * (0.5)^{n-l}$

- We define $Z_k$ as the event that a bot is successful in the validation phase provided that k Sybil flips were assigned.
  $Pr[Z_k] = Pr[X = k] * Pr[Y_{6-k} >= (6 - k - 1)]$

- Since $Z_k$ for $k \in 0, .., 6$ are exclusive events, we can compute the probability of the bot being successful as $Pr[Z] = sum(Pr[Z_k])$
- E[new sybil nodes] = $I * P[Z]$ (I = Available invitation in that round).
- E[sybil nodes at the end of round i] = M + E[new sybil nodes]
- Following an iterative approach we can compute E[sybil nodes for round i+1]

#### A.1.4 Attacker 3

This attackers shares aim and capabilities with the previous attacker but uses a slightly different alogrithm to increase to probability of succeed in every round.

**Aim**

Taking over the network (in this case controlling 33% of the network) by reinvesting the benefit from the network.

**Capabilities**

- The attacker has enough investment capacity to pay for the first nodes $N_0$ entering the network.
- The attacker has an automatic process (bot) that can access a Sybil flip database (containing Sybil flips for a given round).
- The protocol execution time is negligible compared to the short ceremony duration. In practice, the bots and minions have enough time to solve the short validation phase after running the protocol.

**Protocol**

**Initial rounds:**

- The attacker initially hires $N_0$ minions (by making an initial investment).
- The minions will be assigned to nodes that will join the network and (after some initial rounds) that will be verified by the network (obtaining Human status).
- In every round, the minions are paid the UBI for that node (including ceremony rewards and mining rewards).
- After those initial rounds, the amount of available minions is $N_0$ ($M = N_0$), i.e. we can pay as many minions as validated Sybil nodes.

**Subsequent rounds:**

Setup phase:

- The available amount of minions $M$.
- The nodes control by the attacker $Sybil\_nodes = M$
- For each minion the attacker has to submit $f$ flips. Those flips are shared with the bot DB.
- The attacker gets $I$ invitations. The attacker activates those invitations
- The amount of sybil candidates in this round is $Sybil\_Cand = Sybil\_nodes + I$. We assign each candidate and index i.e. $i = \{1, .., C\}$.

- It is important for this attacker to distinguish between *Sybil_nodes* (already verified nodes) and $I$ (new candidates in this round).

Validation phase:

```
Validated_nodes = 0
Available_minions = M
For every c_i in Sybil_nodes:
    Validated_nodes++
    If $c_i.flips$ contain 5 or 6 Sybil flips:
        The bot can safely solve the validation ceremony for that node
        This node is NOT assigned one minion
    Else:
        This node is assigned one minion, then
        Available_minions--;

#We use available minions to solve fresh candidates
Validated_nodes += Available_minions
#For the rest of them we try with LoR-Oracle
Left_inv = I - Available_minions

For every c_i in Left_inv:
    If simulate_bot(c_i) == SUCCESS
        Validated_node++

def simulate_bot(c_i):
    solved_flips = 0
    For c_i.flips:
        If flip is a Sybil flip:
            solved_flips++
        Else
            if toss_coin() == 'HEADS':
                solved_flips++
    If solved_flips >= 5
        return SUCCESS
    Else
        return FAIL
```

At the end of the validation phase:

- Sybil_nodes = Validated_nodes
- The attacker invests the network revenue in that epoch to hire minions for the next round, thus $M = Sybil\_nodes$

**Probability discussion**

The expectation computations merge lucky nodes and successful bots probabilities:

- X being the random variable which indicates how many Bots flips are assigned to one node (out of 6):
- $Pr(X = k) = \binom{6}{k} * sybil\_ratio^k * (1 - Sybil\_ratio)^{6-k}$ (with $k \in 1, .., 6$)
- Y_n being the legitimate flips correctly answered by one Bot node (out of n flips):
- $Pr[Y_n = l] = \binom{n}{l} * 0.5^l * (0.5)^{n-l}$
- We define $Z_k$ as the event that a bot is successful in the validation phase provided that k Sybil flips were assigned. $Pr[Z_k] = Pr[X = k] * Pr[Y_{6-k} >= (6 - k - 1)]$
- since $Z_k$ for $k \in 1, .., 6$ are exclusive events, we can compute the probability of the bot being successful as $Pr[Z] = sum(Pr[Z_k])$

- E[Sybil nodes $|X >= 5] = M * P(X >= 5)$ (M = sybil nodes at the beggining of the round).
- E[successful bots] = $(I - E[Sybilnodes|X >= 5]) * P[Z]$ (I = Available invitation in that round).
- E[new sybil nodes] = E[Sybil nodes $|X >= 5]$ + E[successful bots].
- E[sybil nodes at the end of round i] = M + E[new sybil nodes]

### A.1.5 Attacker 4

This attackers shares aim and capabilities with the previous attacker but uses a different algorithm to increase to probability of succeed in every round.

**Aim**

Taking over the network (in this case controlling 33% of the network) by reinvesting the benefit from the network.

**Capabilities**

- The attacker has enough investment capacity to pay for the first nodes $N_0$ entering the network.
- The attacker has an automatic process (bot) that can access a Sybil flip database (containing Sybil flips for a given round).
- The protocol execution time is negligible compared to the short ceremony duration. In practice, the bots and minions have enough time to solve the short validation phase after running the protocol.

**Protocol**

**Initial rounds:**

- The attacker initially hires $N_0$ minions (by making an initial investment).
- The minions will be assigned to nodes that will join the network and (after some initial rounds) that will be verified by the network (obtaining Human status).
- In every round, the minions are paid the UBI for that node (including ceremony rewards and mining rewards).
- After those initial rounds, the amount of available minions is $N_0$ ($M = N_0$), i.e. we can pay as many minions as validated Sybil nodes.

**Subsequent rounds:**

Setup phase:

- The available amount of minions $M$.
- The nodes control by the attacker $Sybil\_nodes = M$
- For each minion the attacker has to submit $f$ flips. Those flips are shared with the bot DB.
- The attacker gets $I$ invitations. The attacker activates those invitations
- The amount of sybil candidates in this round is $Sybil\_Cand = Sybil\_nodes + I$. We assign each candidate and index i.e. $i = \{1, .., C\}$.
- This attacker, in contrast to the previous one, will assign the available minions to those Sybil nodes which have been assigned less Sybil flips (i.e. the nodes for which validating using a bot would be harder). In other words, we are assigning bots to those nodes which have been assigned more Sybil flips.

Validation phase:

```
Validated_nodes = 0
Available_minions = M
For every c_i in ordered_by_flips_assigned(Sybil_Cand):
    If $Available_minions$ > 0:
        This node is assigned one minion
        Validated_node++
        Available_minion--
    Else:
       If simulate_bot(c_i) == SUCCESS:
        Validated_node++

def simulate_bot(c_i):
    solved_flips = 0
    For c_i.flips:
        If flip is a Sybil flip:
            solved_flips++
        Else
            if toss_coin() == 'HEADS':
                solved_flips++
    If solved_flips >= 5
        return SUCCESS
    Else
        return FAIL
```

At the end of the validation phase:

- Sybil_nodes = Validated_nodes
- The attacker invests the network revenue in that epoch to hire minions for the next round, thus $M = Sybil\_nodes$

## A.2 Simulators vs Probability computations

This section contains some arbitrary experiments conducted using both, the simulator and the probability computations which show consistent outcomes.

### Experiment A

- Attacker 1
- Uses Network Model 1
- Inital network size = 1000
- Inital minions $N_0 = 100$
- Invitation ratio = 1.0

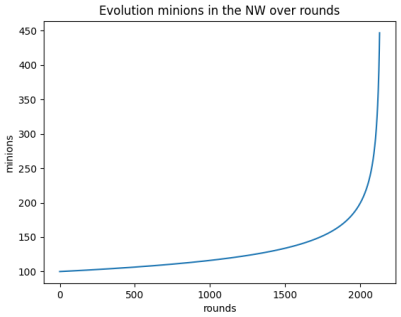
*Results over 15 simulation runs:*

```
Median: 2359.0
Std: 329.7133030713535
Avg:  2390.153846153846
```

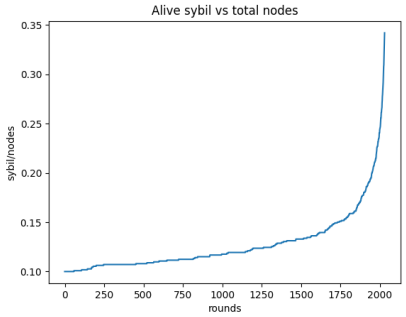
The Figure 20 compares the simulator and the probability computations for the experiment A.

### Experiment B

- Attacker 2




(a) Probability computations



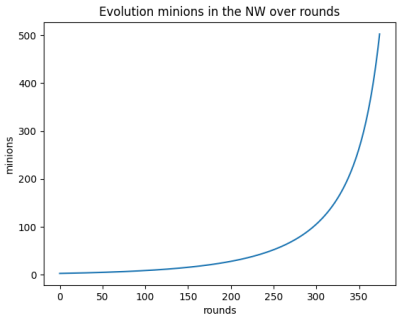

(b) One full simulator run

Figure 20: Experiment A

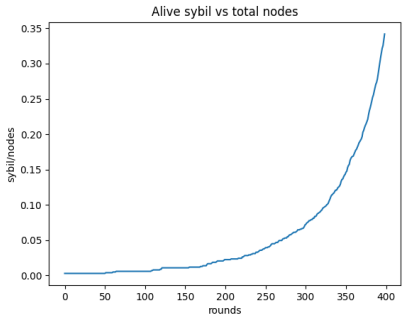
- Uses Network Model 1
- Initial network size = 1000
- Initial minions = 3
- Invitation ratio = 0.1

*Results over 15 simulation runs:*

```
Median: 375.5
Std: 31.589357385043463
Avg:  371.4736842105263
```




(a) Probability computations




(b) One full simulator run

Figure 21: Experiment B

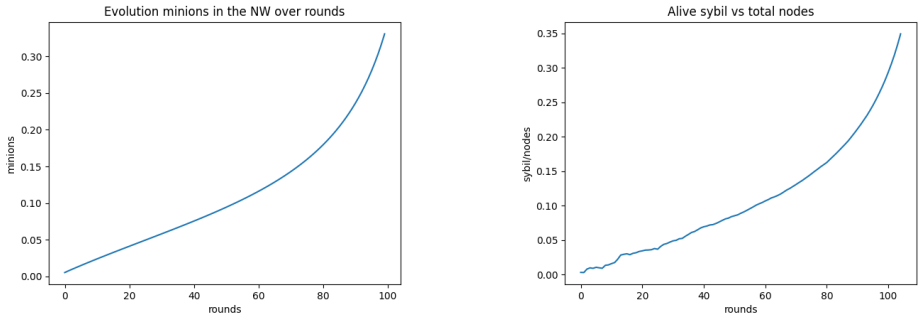
The Figure 21 compares the simulator and the probability computations for the experiment B.

**Experiment C**

- Attacker 3
- Uses Network Model 2
- Initial network size = 333 (222 validated nodes + 111 newbies)
- Initial minions = 1
- Invitation ratio = 0.5
- Persuaded ratio = 0.066

*Results over 15 simulation runs:*

```
Median: 99.0
Std: 3.499841266241783
Avg:  98.85714285714286
```




(a) Probabilistic computation

(b) one lw-simulator run

Figure 22: Experiment C

The Figure 22 compares the simulator and the probability computations for the experiment C.

# References


[1] M. Borge, E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, and B. Ford, "Proof-of-personhood: Redemocratizing permissionless cryptocurrencies."

[2] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system." [Online]. Available: www.bitcoin.org

[3] V. Buterin and V. Griffith, "Casper the friendly finality gadget," 10 2017. [Online]. Available: http://arxiv.org/abs/1710.09437

[4] "Idena website." [Online]. Available: https://www.idena.io/

[5] Manuel, H. N. J., L. J. von Ahn Luis, and Blum, "Captcha: Using hard ai problems for security," E. Biham, Ed. Springer Berlin Heidelberg, 2003, pp. 294–311.

[6] "Idena telegram channel." [Online]. Available: https://t.me/IdenaNetworkPublic

[7] J. R. Douceur, "The sybil attack," Frans, R. A. D. Peter, and Kaashoek, Eds. Springer Berlin Heidelberg, 2002, pp. 251–260.

[8] "Amazon mechanical turk." [Online]. Available: https://www.mturk.com/

[9] B. Ford, "Identity and personhood in digital democracy: Evaluating inclusion, equality, security, and privacy in pseudonym parties and other proofs of personhood," 11 2020. [Online]. Available: http://arxiv.org/abs/2011.02412

[10] D. Siddarth, S. Ivliev, S. Siri, and P. Berman, "Who watches the watchmen? a review of subjective approaches for sybil-resistance in proof of personhood protocols," 2020.

[11] "Idena whitepaper." [Online]. Available: https://docs.idena.io/docs/wp/summary/

[12] "Idena faq." [Online]. Available: https://www.idena.io/faq#faq-attacks-1

[13] "Idena github public repository." [Online]. Available: https://github.com/idena-network

[14] S. Micali, S. Vadhan, and M. Rabin, "Verifiable random functions," in *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, ser. FOCS '99. USA: IEEE Computer Society, 1999, p. 120.

[15] “Idena stats website.” [Online]. Available: https://idena.today/validation.php

[16] “Brightid: Universal proof of uniqueness.” [Online]. Available: https://www.brightid.org/whitepaper

[17] C. Lesaege, W. George, and F. Ast, “Kleros,” 2020. [Online]. Available: https://court.kleros.io