Incentivizing Geographic Diversity for Decentralized Systems

Marc Roeschlin[1], Evangelos Markakis[3,1], Raghav Bhaskar[1], and Aggelos Kiayias[2,1]

[1] Input Output firstname.lastname@iohk.io
[2] The University of Edinburgh, Edinburgh, UK
[3] Athens University of Economics and Business, Athens, Greece

Abstract. Permissionless Decentralized networks, such as blockchains, are typified by self-determined participation. Unfortunately, this has resulted in lack of geographic diversity in several blockchains due to benefits emanating from network proximity between nodes and the higher availability of computing infrastructure in certain areas. Lack of diversity in the resulting network can make it susceptible to geopolitical events, blockchain or cryptocurrency-adverse law-making, and natural disasters. While there exists a growing body of work in verifiable localization in distributed systems, very little exists on mechanisms promoting geographic diversity in distributed systems. Our work sets out to initiate the study of the incentivization of geographic diversity in permissionless distributed systems. We design a family of mechanisms that incentivize network nodes to truthfully declare and diversify their locations. In particular, we provide a game theoretic analysis to derive the conditions under which truthful location reporting is an equilibrium. The conditions relate the offered rewards (for geo-diversity) and the success probability of the underlying localization protocol to detect falsely claimed locations. Our proposed mechanisms assume an underlying secure node localization protocol based solely on round-trip times (RTT) measurements from participants of the protocol. We initiate a formal model to reason about such localization protocols and identify network topologies that are ideal for resisting location spoofing attempts. We evaluate effectiveness of our incentive mechanisms in different scenarios of node placement and underlying network structure. Our validation is based on two RTT data sets we use to derive maximal spoofing distance and attack success rates that adversarial nodes can achieve when operating alone or in collusion with other nodes.

Keywords: Game theory for security, privacy, and blockchain · Blockchain applications · Network and distributed system security

1 Introduction

Decentralized systems such as blockchains typically rely on multiple networked nodes that are connected via the Internet and ensure the immutability and liveness of an underlying ledger. In Bitcoin [23], a permissionless protocol, the number of block-producing nodes varies, and generally, anyone can participate in block-production by running the mining software on suitable equipment. Thus, in the permissionless

setting, node selection is driven only by self-interest without any central coordination. Furthermore, some proof-of-stake blockchains–even when permissionless–may implement mechanisms that limit committee sizes, i.e., the number of nodes dedicated to consensus, block-production, synchronization tasks and data dissemination. For example, the synchronization committee in Ethereum is fixed to 512 nodes [12], and Cosmos has 200 active validators [3].

As a result, the protocol may depend on a small set of nodes without any guarantees as to where the nodes are located geographically. In permissioned protocols, on the other hand, the set of nodes running the protocol can be suitably curated to meet various objectives. These considerations become relevant in terms of the decentralization of the resulting system.

We argue that a decentralized system should also be decentralized in the geographical sense. Moreover, the Internet topology does not span the globe in a uniform manner; hence a set of nodes that is geographically diverse might not result in a uniform set of occupied vertices on the Internet graph, and vice versa. Geographic diversity is also often at odds with economical incentives for node operators: a rational operator aims to reduce the cost for computing power and bandwidth, selecting data centers in strategic locations with high-speed Internet access. Especially for block production nodes, a location that is well connected and offers low-latency communication can make a difference in terms of rewards obtained. Depending on the consensus mechanism, differences in latency and throughput can lead to block races, where a node is more likely to be successful in advancing the chain by proposing a new block if the node is well connected and the generated block reaches the other nodes in time. Observations in the wild confirm these conjectures, and they show, for example, an agglomeration of nodes around important Internet network hubs, such as Frankfurt [9,8].

A consequence of geographic non-diversity is that geopolitical issues, censorship, natural disasters, or other events, even if focused on one country or area, can affect a considerable proportion of a distributed system. Therefore, it would be sensible for blockchain system communities to strive to make their systems more resilient by ensuring a degree of geographic dispersal. However, this poses a difficult conundrum: how to counter network centralization incentives in a way that the system can span to cover remote and less connected regions where it would be required for operators to tolerate economically less favorable locations?

Contributions. We address these considerations by presenting a first study of incentive mechanisms that promote geographic diversity in decentralized systems. We design a reward scheme for node operators that integrates a geography-based reward and promotes geographic diversity that is compatible with any underlying decentralized blockchain system that supports smart contracts.

To ensure that node operators cannot claim a geographic bonus they are not eligible for, we devise a way for the system to verify node locations. We develop a formal model to reason about geolocation verification protocols and identify the salient graph-theoretic properties of the underlying network to facilitate effective verification. The basis of verification is the fact that communication speeds are upper bounded in the physical layer and nodes can only spoof their location to be further but not closer to a verification point that sends ping packets and measures round trip times

(RTTs). It follows that an appropriately distributed set of nodes can catch deviations if a node tries to spoof their location.

Given our model, we develop a mechanism that uses RTT measurements to promote truthful reporting of the operators' locations. A key feature of the mechanism is a distributed location verification algorithm. Using deposits and rewards for successful completion of the geolocation sub-protocol, we create counter incentives for nodes that misreport their location. We provide a game-theoretic analysis of our mechanism, establishing that truthful reporting is a Nash equilibrium. We then analyze the mechanism in the presence of coalitions of players acting together, and we derive sufficient conditions for collusion-resistance.

We conclude with simulations using two real-world datasets that provide network locations and RTT measurements. We determine the maximum spoofing distance that the system can tolerate in the presence of collusions, for both uniform locations and targeted node placement. We estimate the probability of detection of malicious nodes as well as false positive rates in the setting where coalitions of players act together. Our experiments show the practical relevance of our approach and the possibility to deploy our mechanism in real-world decentralized systems.

2 Background and Related Work

Our approach is fundamentally based on *geolocation* and *geolocalization* on the Internet. In particular, we focus on delay-based geolocation which, compared to global positioning systems (GPS), requires neither a specialized radio frequency infrastructure nor any third party that emits a trusted broadcast signal. It utilizes the existing network connection and is uniquely suited for the localization of nodes in a decentralized system, such as block-producing nodes in a blockchain.

Delay-based Location Verification. There are many approaches that measure the delay a message experiences when traversing the Internet, see, e.g., [24,2,14,16,15,13], and the survey in [29]. The network delays between a host and so-called landmarks or anchors with known locations are translated to distances by means of an empirical speed fit, relying on the assumption that the propagation speed in electrical conductors and optical cables is reasonably constant. A rough estimate of a packet's speed on the Internet is around 66% of the speed of light. Although jitter and routing add noise, the geographic (Great Circle) distance correlates well with measured delays. There have been many works on how to map delays to distances more accurately; see, e.g., [18,25,11,19,20,10].

Regardless of the exact distance calculation, most delay-based *geolocation protocols* are largely based on the following main steps:

1. A node makes a location claim (in terms of coordinates) and announces its IP address (if not already known)
2. The landmarks/anchors conduct a series of measurements involving the IP address of the node in question. Often, landmarks probe the node by sending one or more pings and record the time the request was sent alongside the arrival time of the response.

3. Approximate distances are computed based on the delays, which in turn are calculated from the transmission and arrival times of the messages
4. The distances obtained are used in trilateration/multilateration to determine the unknown position or area of the node.
5. If the unknown position is "close enough" to the claimed location or area, the location claim is verified.

Distance Manipulation Attacks. Most delay-based geolocalization mechanisms are vulnerable to distance manipulation that can lead to location misreporting and spoofing [1]. If geolocalization is based on standard ICMP utilities, an adversary can misrepresent their location by affecting the creation and parsing of ICMP packets [1] where sender and responder can independently and almost arbitrarily shorten or increase measured round-trip times (RTTs). Paired with the ability to precompute hypothetical delays between landmarks and the contrived location, an adversary can spoof their location to the desired coordinates by delaying or advancing the respective ping measurements.

Security of RTT Measurements and Secure Positioning. Standard ICMP and TCP pinging methods do not feature cryptographic linking of the request with the response, and thus distance manipulation is feasible. The initiator of a ping measurement can encode an arbitrary timestamp for transmission of the ping, and therefore the correctness of the transmission time cannot be verified. This extends to the reception (and transmission time) at the responder. It is impossible to determine whether the responding node delayed or sent the response prematurely. A way to turn RTT measurements into a distance-bounding-like protocol is to introduce a nonce in the variable-length data field of the ping to ensure that the request cannot be deflated[4], and thus an adversary cannot fake a shorter distance [17] (inflating is still feasible).

We consider this more secure variant of RTT measurements which is applicable provided that the responder is restricted to a single position, i.e., cryptographic material is not shared across multiple locations. Prior work in [6] investigates the more general problem of secure positioning in the presence of a distributed malicious prover; however, it does not address the case of dishonest or colluding verifiers, which is a fundamental concern in decentralized settings.

Geometric Constraints. Constraint-based geolocation (CBG) [14] transforms delay measurements to distance constraints and uses multilateration to infer the location of the target host. The feasible area for the location estimate is formed by calculating the intersection of k circles that are centered around each of the k landmarks, where the circles themselves are defined by the radii that correspond to the maximum possible distance between the target host and the respective landmark. A modification of this technique is used in [17] and [26] to compute the confidence area of a location estimate. A more general framework for contraint-based localization is presented in [28], which determines the estimated target location as a region bounded by a set of Bézier curves allowing for positive and negative constraints.

Decentralization. As CBG [14] and Octant [28] rely on trusted landmarks, it is uncertain whether such schemes can find adoption in a decentralized context. The

[4] To rule out man-in-the-middle attacks, the nonce can be used as part of a challenge-response type mechanism; e.g., the responding node must apply a keyed hash function to the nonce.

first approach to extend delay-based geolocation to a fully decentralized setup is *Verloc* in [17], which performs measurements with a randomly chosen subset of nodes acting as challengers instead of predefined landmarks. Using distributed and verifiable randomness (VRF), VerLoc derives symmetric measurement sets for every node. After a series of ping messages, every node announces their measured RTTs on a blockchain. Each node can access the entire set of measured RTTs and verify the location claims of other nodes by running Newton gradient descent on a coordinate grid based on the latencies posted on-chain. If the claimed location of a node is statistically close to the estimated location, the claim is verified. BFT-PoLoc [26] presents a similar approach, but splits the process into a proof-of-internet-geometry mechanism and a proof-of-location protocol. The proof-of-internet-geometry mechanism calibrates the delay-to-distance mapping for every challenger in a Byzantine fortified manner before the target location is verified by probing it with ping requests. None of these works deal with the issue of incentivizing geographic diversity.

Game-theoretic Approaches. Although not directly related, the work in [4] is conceptually interesting, as it presents security games for node localization through verifiable multilateration in wireless sensor networks (WSNs). The authors define the properties of verifiable multilateration as a non-cooperative two-player game where one player places the verifiers in the monitored area and the other player controls a malicious node. While the first player's goal is to accurately locate any (malicious) node, the second player tries to evade localization and possibly report false locations.

The paper in [22] also considers a game-theoretic view, but differs from our work as the focus is on verifying the location of a single source, by appropriately incentivizing a set of *observers*. The authors analyze proof-of-location as a signal network application and define a notion called *source identifiability*, which is a necessary condition for the existence of a mechanism where truthful signal reporting is a strict equilibrium. Interestingly, their analysis shows that the geometric constraints of localization in a two-dimensional plane imply that the source's location can be truthfully elicited only if it lies inside the convex hull of the observers.

3 Distributed Location Verification – A Formal Model

We now present our framework for secure node localization in network graphs based on secure RTT measurements. We model the network where we wish to run the localization protocol as a weighted graph where the edge weight represents the latency ('true' RTT) between the two nodes. A participant in the localization protocol can at best be 'localized' to one of the nodes of this graph. While an honest participant in the localization protocol will claim the closest node of the graph as its location, a malicious participant may claim any node as its location. The goal of the localization protocol is to detect malicious claims based on RTTs reported by all the participating nodes. Note that in the framework presented here, we assume at most one dishonest node. In Sections 4 and 5, we discuss incentive-based extensions for the case of multiple malicious nodes, who may collude with each other. Extending the formal model presented in this section to the collusion case remains an open problem.

Our graph-based modeling of the network differs from existing approaches in literature, where nodes can claim a set of coordinates as their location. A key motivation for doing this is to be agnostic to the underlying RTT measurement procedure and thus to be widely applicable. Another goal is to study the possible connection of the localization problem with the rich literature of graph-theory. In fact, we establish an important connection between the Geodetic set of a graph [5,7] and the maximum distance that a malicious node can spoof its location by (discussed later). For a more detailed comparison between coordinate-based and our graph-theoretic approach, we refer the reader to Appendix C.

Network Graph. We start with a weighted and undirected graph $G=(V,E)$ that represents the entire network. The weights represent latencies or distances. We assume that latencies are bidirectional. V represents all possible vertices/locations where a node can be located. $A \subseteq V$ represents the nodes participating in the localization protocol. A is a non-empty subset of V with cardinality at least 2. $A_i = A \setminus \{i\}$ is the set of nodes that measure RTTs to node $i \in A$, i.e., the reference set of node i. If the system is comprised of n nodes in total, then $\|A\| = n$ and we assume $\|A_{\neq i}\| = n-1$, i.e., no two nodes are ever at the same vertex.

(δ,A)-further-off Relation. We now present a simple characterization for a participant in any RTT based localization protocol to successfully claim a location other than its true location (*i.e.* to spoof a location) in a given network graph G. The idea is simple: a participant (on her own) can only delay RTTs reported by other participants but cannot shorten them. Thus, if A is the set of participants reporting RTTs to $i \in V \setminus A$, then node i can spoof a location $j \in V \setminus A$ if each participant in A is at some location k such that the RTT between i and k is already smaller (or equal to) than the RTT between each j and k. Thus, there is no need for i to have to shorten any RTT measured to it by others. In such as case, we say that node location j is further-off from i and can be spoofed by i. We capture this formally by defining a *further-off* relation. In all this, we assume no collusion between the participants of the localization protocol.

Notation. In a graph $G=(V,E)$, the length of the shortest path between vertices $i,j \in V$ is denoted by $d(i,j) \in \mathbb{R}_{\geq 0}$. It holds that $d(i,j) = d(j,i)$, and $d(i,j) > 0$ if $i \neq j$, and $d(i,i) = 0$. Also, for any three vertices i, j and w, it holds that $d(i,j) \leq d(i,w) + d(w,j)$.

Definition. The (δ,A)-*further-off* relation denoted $\preceq_A^\delta$ is parametrized on set $A \subseteq V$ and distance $\delta \in \mathbb{R}_{\geq 0}$. Any vertex $j \in V \setminus A$ is (δ,A)-*further off* from vertex $i \in V \setminus A$ iff

$$i \preceq_A^\delta j \;:\; d(i,k) \leq d(j,k) \;\wedge\; d(i,j) \geq \delta \quad \forall k \in A$$

Here we have put the additional constraint that j is at least δ away from i, signifying that the *further-off* is interesting when node i spoofs a location j at least δ away, as spoofing to very nearby locations is trivial.

(A,δ)-Geostable Graph. Using the notion of the *further-off* relation as defined above, we describe our ideal graph, which we call a Geostable graph. Essentially a given graph G is (A,δ)-Geostable for a set of participants A with parameter δ, if for any new participant $i \in V \setminus A$, there is always a participant $k \in A$ which is 'closer' to any 'spoofing' location $j \in V \setminus A$ than i. Here as before, the spoofing location j is at

least 'distance' δ away from i. Thus, no $i,j \in V \setminus A$, are in a (δ,A)-further-off relation, i.e., $\forall_{\substack{i,j \in V \setminus A \\ i \neq j}} \neg\big(i \preceq_A^{\delta} j \big)$, which is equivalent to

$$\forall_{\substack{i,j \in V \setminus A \\ i \neq j}} \exists_{k \in A} \; [d(i,k) > d(j,k) \vee d(i,j) < \delta] = \forall_{\substack{i,j \in V \setminus A \\ i \neq j \; \wedge \; d(i,j) \geq \delta}} \exists_{k \in A} \; d(i,k) > d(j,k)$$

Definition. A graph $G=(V,E)$ is (A,δ)-*Geostable* (with respect to set A) iff

$$\forall_{\substack{i,j \in V \setminus A \\ i \neq j \; \wedge \; d(i,j) \geq \delta}} \exists_{k \in A} \; d(i,k) > d(j,k)$$

Thus the notion of Geostability tells us that if we can find a set $A \subseteq V$ in a graph G, such that no node in $V \setminus A$ is in a (δ,A)-*further-off* relation with $j \in V \setminus A$, then no node can spoof its location to a node location greater than (or equal to) δ away. This is so because a simple algorithm that looks at mismatches between the reported RTTs and the graph weights, can detect the spoofing attempt. Note the δ (we call it the maximum spoofing distance) in the above definition is for the case when the spoofing node is on its own (no collusion with any other node). Later in Section 5 we present an algorithm to compute the maximum spoofing distance when upto t nodes may be colluding and the localization algorithm is more relaxed, in the sense, that it allows upto a threshold number of mismatches (k) between the reported RTTs and the graph weights. Also, note that the Geostability notion does not consider any latency variation in the network. But since δ is the maximum spoofing distance, reasonable jitter is unlikely to impact it (we show experiments in Appendix G to establish this). Further, due to lack of space, we show in appendix A, an interesting connection between Geostability and the (strong) Geodetic set of a graph. In particular, we show that if the set of nodes participating in the localization protocol (A) are located at the Geodetic set locations of the graph, no node can spoof its location to any other node in $V \setminus A$. As mentioned earlier, extending the above notions and definitions to the case of collusion is an open problem, but we present an incentive-based approach to handle collusion in the following section.

4 Generic Model of Games for Truthful Reporting and Geo-diversity

In this section, we develop a game-theoretic framework to prevent spoofing in distributed localisation protocols, even when a set of participating nodes may be colluding. We present a simple and abstract model that can serve as a starting point for studying rewards and incentives when it comes to promoting geo diversity and to discouraging users from trying to spoof their location.

A 2-tier Reward System. The simplest possible reward scheme that a system designer could use would be a tiered system as follows: Tier 1 contains all the operators who were not caught spoofing their location. In particular, a reward r is given to these operators for playing honestly. Tier 2 is defined by determining a priori a set of designated locations where we want to promote further participation of nodes. Then for every active operator who was not caught spoofing her location, and whose

reported location belongs to the designated regions, she receives an additional amount g as a geo-dependent bonus.

We note that if we want to make the protocol more budget-efficient, we could set $r=0$, i.e., we pay only the operators in the designated regions. Moreover, the operators also receive rewards from block production. We assume these are not affected by the game we consider here, i.e., whether they spoof their location or not, they will continue producing and validating blocks, and hence receiving such rewards. Therefore, we will not incorporate them in the game of interest here. The game proceeds as follows:

- In Phase 1, the protocol asks all operators to report their location and to also submit a deposit $d \geq 0$.
- In Phase 2, the protocol runs an algorithm to determine whether the operators are truthful. We treat this as a black box here. This algorithm can possibly ask for further information from the nodes, such as RTT times.
- If at any point during Phase 1 or Phase 2, a node does not abide by the protocol, i.e., does not respond to whatever information the protocol asks, she loses her deposit.
- Finally, in Phase 3, the protocol issues payments as follows: Anyone who was caught lying does not receive any payment. All other players receive back the deposit d and a reward r. Among these players, the operators who are in designated regions also receive the extra geographic bonus, g.

Strategy Space. At this level of description, we abstract away all the possible moves of a player (which also depend on how exactly Phase 2 is implemented) and we view this as a standard normal-form game, where each player has to decide independently of the other players, among 2 actions: either she can decide to play honestly (denoted by H), meaning that she reports truthfully and makes no attempt to fake her location, or she can try to deviate, misreport and/or cheat on her claimed location (denoted by C). Technically, cheating can occur in many ways depending on how the protocol tries to verify the locations, but we feel it is insightful to start with this binary choice for now. The motive for cheating here is to pretend that a node belongs to one of the designated regions so as to receive the additional bonus of g. In Section 4.1, we also discuss the possibility of collusions among players.

In order to analyze any verification protocol in line with the above model, we define first some relevant parameters. Fix a player i, and let $\mathbf{x}$ be a strategy profile for the other players, i.e., $\mathbf{x}$ contains a strategy choice for all players other than i, $\mathbf{x} \in \{H,C\}^{n-1}$, where n is the total number of operators. Given that a verification protocol is expected to have randomized decisions, we let $p_s(i,\mathbf{x})$ be the conditional probability of i getting caught, when she tries to spoof her location, and given also that the other operators behave according to $\mathbf{x}$. Analogously, we let $p_h(i,\mathbf{x})$ be the conditional probability of i being marked as dishonest/malicious by the protocol, conditioned that she abided honestly by the protocol rules and the other operators play $\mathbf{x}$. The latter can occur when the protocol could produce false positives. Naturally, we can expect that $p_h(i,\mathbf{x}) << p_s(i,\mathbf{x})$. Moreover, $p_h(i,\mathbf{x})=0$ if a protocol has no false positives.

The essence of the next theorem is that if we can estimate the probabilities $p_s(\cdot)$ and $p_h(\cdot)$ in a given protocol, then we could tune the reward parameters r, g and the deposit d, so that playing honestly is the most preferable choice for each player.

Theorem 1. *Fix a player i and let $\mathbf{x} \in \{H,C\}^{n-1}$ be the strategy profile of the remaining players. If i is not located in the designated regions, then playing honestly is better for i than deviating, when the geographic bonus is upper-bounded as follows*

$$g \leq \frac{p_s(i,\mathbf{x}) - p_h(i,\mathbf{x})}{1 - p_s(i,\mathbf{x})} \cdot (r+d)$$

Proof. Consider such a player i. When i plays honestly, and the remaining players play according to $\mathbf{x}$, then her expected payoff is equal to $(1-p_h(i,\mathbf{x}))r + p_h(i,\mathbf{x})(-d)$. This is so, because i will get her deposit back if she is not marked as dishonest by the protocol, which occurs with probability $1-p_h(i,\mathbf{x})$. On the other hand, suppose that such a player i cheats. Then, with probability $p_s(i,\mathbf{x})$, she will have a negative utility of $-d$, since she will lose her deposit, whereas with the remaining probability she gets a total gain of $r+g$. Therefore, her expected utility becomes:

$$(1-p_s(i,\mathbf{x}))(r+g) + p_s(i,\mathbf{x})(-d)$$

In order for honest play to be more preferred, we need the expected payoff under the honest profile to be at least as good as the above formula. Thus, the following needs to hold: $(1-p_h(i,\mathbf{x}))r + p_h(i,\mathbf{x})(-d) \geq (1-p_s(i,\mathbf{x}))(r+g) + p_s(i,\mathbf{x})(-d)$. Simplifying this leads to the claimed condition in the theorem. □

This allows us to conclude when does honest play form a Nash equilibrium of the underlying game. Let $\mathbf{x}^* = (H,H,\cdots,H)$ be the honest profile, where all n players act honestly. Following standard notation, and given a player i, let $\mathbf{x}^*_{-i}$ be the strategy profile of the remaining $n-1$ players acting honestly.

Corollary 1. *The honest profile $\mathbf{x}^* = (H,H,\cdots,H)$ is a Nash equilibrium if for every player i whose real location is in a non-designated region, it holds that*

$$g \leq \frac{p_s(i,\mathbf{x}^*_{-i})}{1 - p_s(i,\mathbf{x}^*_{-i})} \cdot (r+d), \tag{1}$$

where $p_s(i,\mathbf{x}^*_{-i}))$ is the probability of i getting caught when all players are acting honestly.

4.1 Collusion

Coming now to the aspects of collusion, it is conceivable that some operators, say a set S, could try to form a coalition so that some members of S can attempt to spoof their location and receive the extra geographic reward bonus g. We can think of the set S as being decomposed in 2 subsets, $S = S_1 \cup S_2$, where S_1 is the subset of S belonging already to the designated regions and S_2 is the subset of S who want to spoof their location. We will often refer to S_2 as the *spoofers* and to S_1 as the *helpers*.

The reason that S_1 may be willing to participate in such a collusion is because the members of S_2 may offer them some compensation from the extra amount that they will make if they succeed in spoofing their location. Therefore, the members of S_1 may take the risk to help (by reporting say false RTT times towards S_2), even though they will not try to spoof their own location. This gives rise to the following definition.

Definition 1. *Consider a coalition of operators S. We say that the honest profile $(H,H,...,H)$ is resilient to a collusion by S, if the total expected utility of S under the honest profile is at least as big as the total expected utility under deviating.*

For simplicity, we will stick to scenarios where only one set of nodes is trying to collude together and deviate from the protocol in order to gain more profits. Hence, if S is the set of the colluding players, we will consider that the remaining operators are behaving honestly. Ideally, to evaluate the resistance of a protocol to such actions, we would need to estimate the probability that a player gets caught cheating. i.e, we could define $p(i,S)$, as the probability of player i getting caught when the collusion set is S, with $i \in S$, while all other players not in S are being honest. We will provide here a simpler but coarser analysis, assuming that the relevant probabilities depend on the cardinalities of the involved sets and not on the identities of its members.

In particular, let S be a set of colluding players, with $|S|=t$ and suppose $|S_1|=t_h$, $|S_2|=t_s$, so that $t=t_h+t_s$. We continue our analysis by focusing on two different scenarios regarding the success of the collusion. The first one is more suitable for protocols that try to detect deviations by a coalition of players as a whole. The second one captures protocols that try to detect a deviation per individual.

All-or-nothing Detection. Consider the case where the protocol either catches all the spoofers of the collusion or not. We let $p(t,t_s)$ be the probability that a set of t_s spoofers gets caught, when they form a coalition with $t-t_s$ helpers. We assume that when the collusion is detected by the protocol, it penalizes all the spoofers (who then lose their deposit) but not the helpers. Thus, we consider a *lenient* type of protocols that do not hurt the helpers, given that they do not try to misreport their location. The subsequent analysis can be adapted easily for the case where the helpers are also punished.

We can now try to establish some sufficient conditions on the relevant parameters that could avert collusion.

Theorem 2. *The honest profile $\mathbf{x}^*=(H,H,...,H)$ is collusion-resistant to a collusion of size t with t_s spoofers, when*

$$g \leq \frac{p(t,t_s)}{1-p(t,t_s)} \cdot (r+d) \tag{2}$$

Proof. Under the honest profile, the total utility of S is $\sum_{i \in S} u_i(H,H,...,H)$. For each member $i \in S_1$, her utility under honest play equals $(r+g)$ with probability 1. For the members of S_2, it equals r. Therefore, by summing up, we have

$$\sum_{i \in S} u_i(H,H,...,H) = (t-t_s)(r+g)+t_s r \tag{3}$$

If the set S deviates, then with probability $p(t,t_s)$ the members of S_2 lose the deposit d, and do not get any reward. On the other hand, with probability $1-p(t,t_s)$, all members of S gain $r+g$. Hence the expected utility of S would be equal to

$$t(1-p(t,t_s))(r+g)+p(t,t_s)[(t-t_s)(r+g)-t_s d]$$

In order for $\mathbf{x}^*$ to be collusion resistant, we therefore need the last expression to be upper bounded by the expected utility of S under honest play. After carrying out the calculations and some arising simplifications, we obtain the desired inequality. □

Individual Detection. Suppose now that when a set S colludes, the protocol may identify some of the members as spoofers whereas other members may get away with it. In this context, we use the term $p(t,t_s)$ to denote the probability that each spoofer gets caught and penalized, when she belongs to a coalition of t members with t_s spoofers, and when all other players are honest. This means that for each spoofer we have a Bernoulli trial for having her caught. Also, as before, the helpers are not penalized. We provide an analog of Theorem 2 below, where the conditions now get more complex due to the counting of the successful Bernoulli trials.

Theorem 3. *The honest profile* $\mathbf{x}^* = (H,H,...,H)$ *is collusion-resistant to a collusion of size* t *with* t_s *spoofers, when*

$$\sum_{\ell=0}^{t_s} \binom{t_s}{\ell} \cdot p(t,t_s)^{\ell} (1-p(t,t_s))^{t_s-\ell} [-d\ell + (t_s-\ell)(r+g)] \leq r t_s \tag{4}$$

5 Decentralized Localization Game

We now present a decentralized localization game derived from the model proposed in the previous section. In particular, we use a blockchain smart-contract to play the role of the Game Organizer (GO), concretize the exact data shared by the operators, the localization algorithm that is used to detect misbehaving operators (which was treated as a black box in the previous game analysis), and specify the various strategies available to the operators.

Game parameters: Internet graph G, the rewards: r, g and the required deposit d. Also, an upper bound t on the maximum number of colluding nodes in the network is assumed. The localization algorithm has another tweakable parameter k to manage the false positives and false negatives. As a first step, all the game parameters are published on the blockchain and a smart-contract with the above-mentioned parameters embedded in it is launched. As before, the game is divided into phases:

- **First phase:** Operators send their IP address[5], a deposit of $d \geq 0$ coins as well as a commitment (hash function based) to their location (instead of outrightly sending in their location) to the smart contract. Malicious operators may report a location on the internet graph different from their true location, as well as a different IP address, but we assume that ping requests to the claimed IP address is responded to by the operator. Operators can also choose to totally abstain from the game, but they do not affect our game in any way. Note that the commitment hides the location from all other operators and also ensures that the participant cannot open (change) it to a different location later. This way the operators can not adapt their strategy based on other operators' locations/actions. As mentioned in Section 4, our analysis is based on normal-form games, i.e., every operator picks a strategy and then plays the game.

[5] For honest operators, IP addresses may correlate with physical location; however, our approach does not rely on IP-based localization.

Algorithm 1 Localization

Input: Graph $G:=(V=[1,n],E=\{i,j,e[i,j]\}$ s.t. $i,j \in V$, $e[i,j] \geq 0 \wedge e[i,j]=e[j,i])$,
$O \subseteq V$ s.t. $|O|=m$,
$RTTs=\{(i,ip_i,rtt_i[m-1])\}_{i \in O}$, k
Output: Set: Malicious
Steps:
$\forall i \in O$: count$[i]=0$; Malicious $= \{\}$
for $(i \in O, j \in O \setminus \{i\})$ **do**
  if $(rtt_i[j] \neq e[i,j])$ **or** $(rtt_j[i] \neq e[i,j])$ **then**
    count$[i]$++
  end if
end for
if $\exists i \in O$ with count$[i] \geq k$ **then**
  Malicious $\leftarrow$ Malicious $\cup \{i\}$
end if
Output Malicious

Algorithm 2 deltafn(i,O,t,k)

Input: $G = (V, E)$, $O \subseteq V$ s.t. $|O|=m$, $i \in O$, t, k
Output: δ
Steps:
$\delta=0$
for a random choice of set S in $O \setminus \{i\}$ of size $t-1$ **do**
  $H=O \setminus \{S\}$
  for $i' \in V \setminus O$ **do**
    $H' \leftarrow \{j : j \in H,\ e[j,i'] < e[j,i]\}$
    if $|H'| < k$ **then**
      $\delta = \max(\delta, d(i,i'))$
    end if
  end for
end for
return δ

- **Second phase:** All participating nodes read the IP addresses from the smart contract and then perform the $m-1$ RTT measurements, and publish them back to the smart contract as well as send their location (open the commitment) in the graph. As mentioned earlier, a cheating operator may report arbitrary RTTs to the various IP addresses.
- **Third phase:** The smart contract validates that every participant has submitted the required RTT measurements, and then collects the locations submitted by the operators. The smart contract then runs the localization algorithm (Algorithm 1) along with the received location claims (which should match the commitments) to decide on the malicious nodes and the payments and rewards that are to be made.
  - If one or more operators are output as malicious: Forfeit deposit of all malicious nodes. Return deposits of all other non-malicious nodes along with reward r.
  - If the output Malicious set is empty: Each operator gets back the deposit d and a reward r. Further, operators in designated regions receive additional reward g.

Overview of Algorithm 1. The algorithm takes as input the graph G, the operator set O of size m, a $m-1$ array per operator with the measured RTTs and k, which is the number of RTT mismatches that are considered by the localization algorithm to detect misbehaving nodes. k can take any value in $[1,t+1]$ [6], with each value resulting in different false positives and false negatives in misbehaviour detection. For instance, a value of $t+1$ ensures zero false positives (high false negatives as well), as no honest node can have more than t mismatched entries (here we assume that the colluding nodes are not trying to frame any honest node, rather making modifications to RTTs to help the colluding nodes avoid detection). A smaller value of k in contrast leads to lower false negative rates and smaller maximum spoofing distance.

[6] As in the previous section, if we assume that t_s out of the t colluding nodes to be spoofing their location, k can take a value in $[1,t_s+1]$

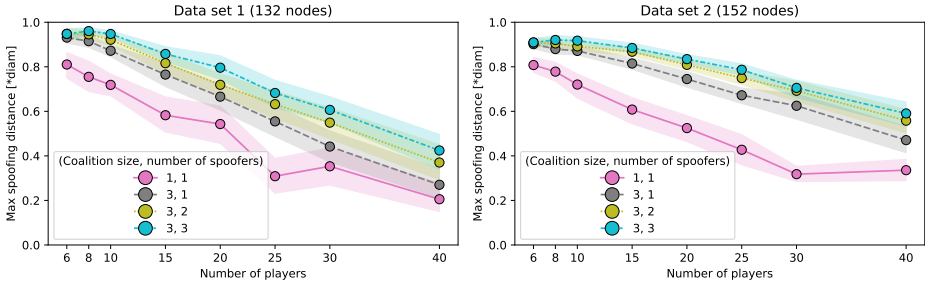


Fig. 1: Maximum spoofing distance (given in terms of diamter of G) for varying numbers of spoofers within a coalition of size 3. Shaded areas indicate the 90% confidence interval.

On the Threshold k. We propose a way for GOs to choose an appropriate value of k. To this end, the GO can use Algorithm 2 to compute $\delta^{i,\mathrm{O},t,k}$ which is the maximum distance any node $i \in \mathrm{O}$ can spoof its location by, when the nodes $(\mathrm{O} \setminus i)$ are reporting RTTs to i, at most $t-1$ of them are in collusion with i, and k is the threshold used in the localization algorithm. Further $\delta^{G,\mathrm{O},t,k} = max_{i \in \mathrm{O}}$ $\mathsf{deltafn}(i,\mathrm{O},t,k)$. If the value of $\delta^{G,\mathrm{O},t,k}$ is considered too high, then an option may be to decrease k. However, as mentioned before, this also increases the false positive rate.

We further observe that the parameters k and t in Algorithm 1 and 2 constitute a practical bridge between the idealized graph-based formulation in Section 3 and the game-theoretic model in Section 4 that describes a realistic setting where collusion and a limited number of RTT mismatches (k) are allowed. We present some empirical results around k, t and the false positives in Section 6.

6 Simulations

We present some key insights related to the game postulated in Section 5 using simulations on two data sets that contain RTT measurements between hosts in the Internet.

Data Sets. The first data set comprises 132 hosts forming an overlay network. Since the network's architecture and routing are known, the data set includes both RTTs between nodes and edge weights, where the weights represent average delays for the corresponding links. This graph structure enables evaluation of the graph-theoretic aspects discussed in Section 3. The corresponding simulation results are in Appendix E.

The second data set contains 152 hosts with pairwise round-trip time (RTT) measurements, including minimum, maximum, and average delays. It is provided by WonderNetwork [27], a global provider of networking services specializing in performance testing. Both data sets contain the geographic locations of the hosts at city-level precision, which we show in Appendix D.

Maximum Spoofing Distance under Collusion. We examine the maximum spoofing distance (δ) achievable under collusion, using both data sets and assuming a fixed coalition size of three. Figure 1 reports values of $\delta^{G,\mathrm{O},t,k}$ for $t=3$ and $k \in 1,2,3$. $\delta^{G,\mathrm{O},t,k}$ is computed using the function $\mathsf{deltafn}(i,\mathrm{O},t,k)$ for all $i \in \mathrm{O}$ as defined in Algorithm 2.

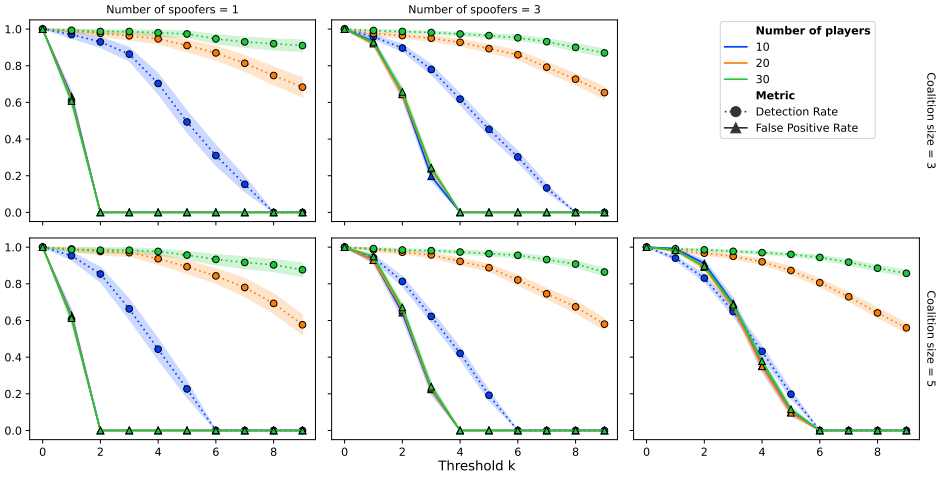
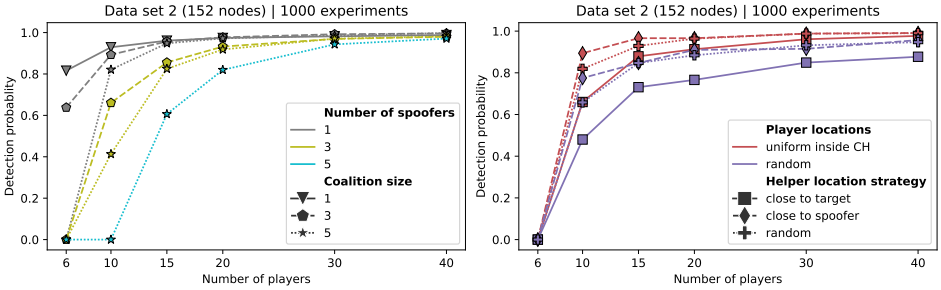


Fig. 2: $p(t,t_s)$ (dotted lines) and $p_h(t,t_s)$ (solid lines) for varying numbers of players, coalition size t, and number of spoofers t_s, based on data set 2 (152 nodes). Shaded bands represent the 90% confidence interval. Player locations are sampled uniformly within the convex hull.

As described before, spoofers falsely report their location and are allowed to manipulate both incoming and outgoing RTT measurements to evade detection. Helpers, by contrast, do not spoof their own location but submit RTTs that favor coalition members. The results in Figure 1 clearly show that the maximum spoofing distance increases with the number of spoofers in the coalition. Maintaining zero false positives requires a higher detection threshold (as discussed in Section 5), which in turn reduces the sensitivity of the system to spoofing attempts, particularly when multiple dishonest players are involved. Compared to the baseline case of a lone spoofer, a coordinated coalition can significantly extend the range over which spoofing remains undetected.

Probability of Malicious Behavior Detection and False Positives. We present insights from our incentivization game from Section 5. We measure the probability that a dishonest player is correctly identified ($p(t,t_s)$) when attempting to spoof their location, and the probability that an honest player is incorrectly flagged as dishonest (false positive), $p_h(t,t_s)$. These metrics are evaluated in various scenarios, including when dishonest players act alone or as part of a coalition. The trade-off between $p(t,t_s)$ and $p_h(t,t_s)$ is controlled by the threshold parameter k in the localization algorithm (Section 5). The results shown here are for data set 2; the results for data set 1 are similar and can be found in Appendix F.

Figure 2 reveals that increasing k consistently reduces $p_h(t,t_s)$. In particular, $p_h(t,t_s)$ drops to zero once k exceeds the number of spoofers. The "helping" players within the coalition contribute favorable RTTs but do not perform spoofing themselves (see Section 4), and therefore do not influence $p_h(t,t_s)$. However, $p(t,t_s)$ decreases with increasing k. This is because Algorithm 1 requires a higher number of honest players located near the spoofed position to register enough measurement inconsistencies and reach the threshold k. The number of both spoofers and helpers affects $p(t,t_s)$; the



(a) Varying number of spoofers and coalition sizes

(b) One spoofer in a coalition of size 5 for different helper locations

Fig. 3: Detection probability ($p(t,t_s)$) for $k=t+1$, i.e., no false positives. Player locations are sampled uniformly inside convex hull. Target locations (during spoofing) are randomly chosen.

more dishonest players are present, the greater the likelihood that a spoofing attempt goes undetected, as the coalition can obscure more RTT mismatches.

Figure 3a shows the average $p(t,t_s)$ when the localization algorithm is configured for low false positives (low $p_h(t,t_s)$), enabling collusion resistance as described in Section 4. We set the threshold k to be the number of spoofers plus one, which ensures that spoofers are only flagged when a sufficient number of honest players register inconsistencies. The results in Figure 3a confirm that the $p(t,t_s)$ increases with the number of players because more players result in more measurement paths and consequently a higher probability of detecting anomalies. Detection becomes more difficult as the coalition size grows. Larger coalitions provide spoofers with more support in manipulating RTT measurements, thereby reducing the odds of detection. A greater number of spoofers further improves the coalitions' chances of evading detection as more spoofers introduce a larger number of RTT mismatches, potentially implicating honest players and lowering the effectiveness of the detection mechanism.

Collusion Strategies. We investigate how the placement of colluding players influences the detection probability of location spoofing. Rather than assuming that helpers are selected randomly and without (geographic) awareness, we consider more strategic adversarial behaviors in which the spoofer chooses helpers located either near their target location or near their actual location.

Figure 3 presents the average detection probability for these three strategies, assuming a coalition of five players, with one acting as the spoofer and four as helpers. The experiments are repeated under two different player location sampling regimes: random selection and uniform distribution within the convex hull. For instance, the solid purple line with square markers in Figure 3 shows the detection probability when player locations are sampled randomly, and the helpers are positioned near the spoofer's target location.

A uniform distribution of players improves the detection probability, but more importantly, there is a decrease in detection probability when helpers are located close to the target. In this configuration, the helpers can report shorter RTTs that are consistent with the spoofed location, as their distance to the target is often shorter than to the spoofer's actual location. This increases the chances of a successful spoofing

attempt. Conversely, when the helpers are placed near the spoofer's true location, the detection probability is largely unaffected. This strategy provides little to no advantage to the attacker, and $p(t,t_s)$ remains comparable to the baseline with random sampling.

Incorporating Jitter and Measurement Noise. Algorithm 1 relies on inequality conditions to detect mismatches in RTTs. In practice, these inequalities may be relaxed to account for variability introduced by routing dynamics and network noise. Our empirical analysis shows that accounting for jitter only has a limited effect on detection rates. Assuming that 10% of the delay is governed by jitter, detection rates decrease by less than 5%. The respective figures can be found in Appendix G.

7 Discussion and Conclusion

We adopted a formal approach to analyze the guarantees that distributed geolocation verification can offer in arbitrary network topologies. We introduced a generic two-tier reward game model designed to incentivize geographic diversity and presented a corresponding localization algorithm tailored to meet the requirements for both Nash equilibrium and collusion resistance. Finally, we evaluated the detection probability and false positive probabilities, the two central metrics underpinning the feasibility of equilibrium and collusion resistance, using two real-world data sets.

Possible Extensions. Our study is a starting point for incentivizing geographic diversity. Currently, players face a binary decision (honest/dishonest), but a multi-tiered system and a richer strategy space would allow for more nuanced decisions, including multiple spoofing targets. Furthermore, future work should consider colluding behaviors where dishonest players undermine honest participants without gaining a direct reward.

Overall, this work advances the theory of distributed location verification and bridges the gap towards practical applications in real-world networks. By facilitating more resilient location verification, our mechanism can help mitigate spoofing attacks and incentivize geographic diversity in distributed systems.

Acknowledgments. This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program.

References

1. Abdou, A., Matrawy, A., van Oorschot, P.C.: Accurate manipulation of delay-based internet geolocation. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. p. 887–898. ASIA CCS '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3052973.3052993, https://doi.org/10.1145/3052973.3052993
2. Arif, M.J., Karunasekera, S., Kulkarni, S., Gunatilaka, A., Ristic, B.: Internet host geolocation using maximum likelihood estimation technique. In: 2010 24th IEEE International Conference on Advanced Information Networking and Applications. pp. 422–429 (2010). https://doi.org/10.1109/AINA.2010.139

3. ATOMScan, Inc.: Validators – ATOMScan. https://atomscan.com/validators (2025), AtomScan: Blockchain explorer for Cosmos validator information
4. Basilico, N., Gatti, N., Monga, M., Sicari, S.: Security games for node localization through verifiable multilateration. IEEE Transactions on Dependable and Secure Computing **11**(1), 72–85 (2014). https://doi.org/10.1109/TDSC.2013.30
5. Brešar, B., Kovše, M., Tepeh, A.: Geodetic Sets in Graphs, pp. 197–218. Birkhäuser Boston, Boston (2011). https://doi.org/10.1007/978-0-8176-4789-6_8, https://doi.org/10.1007/978-0-8176-4789-6_8
6. Chandran, N., Goyal, V., Moriarty, R., Ostrovsky, R.: Position based cryptography. In: Advances in Cryptology – CRYPTO 2009. Lecture Notes in Computer Science, vol. 5677, pp. 391–407. Springer, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03356-8_23, https://doi.org/10.1007/978-3-642-03356-8_23
7. Chartrand, G., Harary, F., Zhang, P.: Geodetic sets in graphs. Discussiones Mathematicae Graph Theory **20**(1), 129–138 (2000)
8. CoinShares: Bitcoin nodes around the world: a look at their distribution & impact (April 2025), https://coinshares.com/ch/insights/knowledge/bitcoin-nodes-around-the-world-a-look-at-their-distribution-impact/, accessed: 2025-05-07
9. Cointelegraph: Bitcoin nodes data: Frankfurt houses the largest city-wide network (2024), https://cointelegraph.com/news/bitcoin-nodes-data-frankfurt-houses-the-largest-city-wide-network, accessed: 2025-05-07
10. Dabek, F., Cox, R., Kaashoek, F., Morris, R.: Vivaldi: a decentralized network coordinate system. SIGCOMM Comput. Commun. Rev. **34**(4), 15–26 (Aug 2004). https://doi.org/10.1145/1030194.1015471, https://doi.org/10.1145/1030194.1015471
11. Dong, Z., Perera, R.D.W., Chandramouli, R., Subbalakshmi, K.P.: Network measurement based modeling and optimization for ip geolocation. Comput. Netw. **56**(1), 85–98 (Jan 2012). https://doi.org/10.1016/j.comnet.2011.08.011, https://doi.org/10.1016/j.comnet.2011.08.011
12. Edgington, B.: Aggregator selection. https://eth2book.info/latest/part2/building_blocks/aggregator/ (Sep 2025), in *Upgrading Ethereum: The Eth2 Book*, Part 2: Technical Overview — The Building Blocks
13. Eriksson, B., Barford, P., Sommers, J., Nowak, R.: A learning-based approach for ip geolocation. In: Krishnamurthy, A., Plattner, B. (eds.) Passive and Active Measurement. pp. 171–180. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
14. Gueye, B., Ziviani, A., Crovella, M., Fdida, S.: Constraint-based geolocation of internet hosts. IEEE/ACM Transactions on Networking **14**(6), 1219–1232 (2006). https://doi.org/10.1109/TNET.2006.886332
15. Hong, A., Li, Y., Zhang, H., Wang, M., An, C., Wang, J.: A cheap and accurate delay-based ip geolocation method using machine learning and looking glass. In: 2023 IFIP Networking Conference (IFIP Networking). pp. 1–9 (2023). https://doi.org/10.23919/IFIPNetworking57963.2023.10186436
16. Katz-Bassett, E., John, J.P., Krishnamurthy, A., Wetherall, D., Anderson, T., Chawathe, Y.: Towards ip geolocation using delay and topology measurements. In: Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement. p. 71–84. IMC '06, Association for Computing Machinery, New York, NY, USA (2006). https://doi.org/10.1145/1177080.1177090, https://doi.org/10.1145/1177080.1177090
17. Kohls, K., Diaz, C.: VerLoc: Verifiable localization in decentralized systems. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 2637–2654. USENIX Association, Boston, MA (Aug 2022), https://www.usenix.org/conference/usenixsecurity22/presentation/kohls

18. Kohls, K., Jansen, K., Rupprecht, D., Holz, T., Pöpper, C.: On the challenges of geographical avoidance for tor. In: NDSS (2019)
19. Krajsa, O., Fojtova, L.: Rtt measurement and its dependence on the real geographical distance. In: 2011 34th International Conference on Telecommunications and Signal Processing (TSP). pp. 231–234 (2011). https://doi.org/10.1109/TSP.2011.6043737
20. Landa, R., Clegg, R.G., Araujo, J.T., Mykoniati, E., Griffin, D., Rio, M.: Measuring the relationships between internet geography and rtt. In: 2013 22nd International Conference on Computer Communication and Networks (ICCCN). pp. 1–7 (2013). https://doi.org/10.1109/ICCCN.2013.6614151
21. Maram, D., Kelkar, M., Bentov, I., Juels, A.: Goat: File geolocation via anchor timestamping. In: Clark, J., Shi, E. (eds.) Financial Cryptography and Data Security. pp. 35–72. Springer Nature Switzerland, Cham (2025)
22. Milionis, J., Ernstberger, J., Bonneau, J., Kominers, S.D., Roughgarden, T.: Incentive-compatible recovery from manipulated signals, with applications to decentralized physical infrastructure (2025), https://arxiv.org/abs/2503.07558
23. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. http://bitcoin.org/bitcoin.pdf (2008)
24. Padamanabban, V.N., Subramanian, L.: Determining the geographic location of internet hosts. In: Proceedings of the 2001 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems. p. 324–325. SIGMETRICS '01, Association for Computing Machinery, New York, NY, USA (2001). https://doi.org/10.1145/378420.378814, https://doi.org/10.1145/378420.378814
25. Padmanabhan, V.N., Subramanian, L.: An investigation of geographic mapping techniques for internet hosts. In: Proceedings of SIGCOMM 2001. p. 173–185. SIGCOMM '01, Association for Computing Machinery, New York, NY, USA (2001). https://doi.org/10.1145/383059.383073, https://doi.org/10.1145/383059.383073
26. Sheng, P., Sevani, V., Rana, R., Tyagi, H., Viswanath, P.: Bft-poloc: A byzantine fortified trigonometric proof of location protocol using internet delays (2024), https://arxiv.org/abs/2403.13230
27. WonderNetwork: A day in the life of the internet. https://wonderproxy.com/blog/a-day-in-the-life-of-the-internet/ (2024), accessed: 2024-10-14
28. Wong, B., Stoyanov, I., Sirer, E.G.: Octant: A comprehensive framework for the geolocalization of internet hosts. In: NSDI. vol. 7, pp. 23–23 (2007)
29. Zilberman, A., Offer, A., Pincu, B., Glickshtein, Y., Kant, R., Brodt, O., Otung, A., Puzis, R., Shabtai, A., Elovici, Y.: A survey on geolocation on the internet. IEEE Communications Surveys & Tutorials pp. 1–1 (2024). https://doi.org/10.1109/COMST.2024.3518398

A Geodetic Sets and Geostability

We recall the notion of Geodetic sets from Graph theory and show that any graph G is Geostable with respect to its Geodetic set.

Notation. The diameter $diam(G)$ of a connected graph is defined by $diam(G) = max_{x,y \in V(G)}\ d(x,y)$. A path $x-y$ of length $d(x,y)$ is called an $x-y$ geodesic. A vertex v is said to lie on an $x-y$ geodesic P if v is an internal vertex of P. The closed interval $I[x,y]$ consists of x, y and all vertices lying on some $x-y$ geodesic of G, while for $A \subseteq V(G)$, $I[A] = \bigcup_{x,y \in A} I(x,y)$.

Definition. If G is a connected graph, then a set A of vertices is a Geodetic set if $I[A]=V(G)$. The minimum cardinality of a Geodetic set is the Geodetic number of G, and is denoted by $g(G)$.

Thus all vertices of a graph G lie on some shortest path between some vertices of a Geodetic set A, or:

$$V(G) := \bigcup_{k,k' \in A} P(k,k')$$

We show that if A is a Geodetic set of G, then G is a $(A,0)$-Geostable graph. This means that if nodes in A are reporting RTTs to any node $i \in V \setminus A$, then i cannot spoof its location to any other node $j \in V \setminus A$ (note $\delta=0$). We now prove the above claim:

Theorem 4. *If A is a* Geodetic *set of graph G, then G is* $(A,0)-$Geostable.

Proof. If A is a Geodetic set of graph G, then all vertices of G and in particular all vertices in $V \setminus A$ lie on some shortest path between two vertices of A. Note, a vertex in $V \setminus A$ may lie on multiple shortest paths between the vertices of A, and if after choosing some selected path for each vertex, the set A continues to remain a Geodetic set, then it is called a *strong* Geodetic set. We will show that A being a *strong* Geodetic set is a sufficient condition for G to be $(A,0)$-Geostable.

First, consider the case, when any two vertices $i,j \in V \setminus A$ lie on the same shortest path $P(k,k')$ connecting $k,k' \in A$. Since $i \neq j$, $d(i,j) > 0$ and this means either $d(i,k) < d(j,k)$ or $d(i,k') < d(j,k')$, depending on whether i lies before j on the path from k to k' or j lies before i on the path from k to k' respectively. Therefore, neither i can spoof j's location nor j can spoof i's location. In general, for any two such nodes $i,j \in V \setminus A$, $\neg\left(i \preceq_A^{\delta} j \right)$ and $\neg\left(j \preceq_A^{\delta} i \right)$.

Now we will show that two nodes in $V \setminus A$ that are not on the same shortest paths, are also unable to spoof each other's location, when the set A is a strong Geodetic set.

We assume that i lies on a unique shortest path between nodes $k_1,k_2 \in A$ and j lies on the unique shortest path between nodes $k_3,k_4 \in A$, where at least $k_1 \neq k_3$ or $k_2 \neq k_4$. Without loss of generality, we will assume $k_1 \neq k_3$ in the analysis presented below. A similar argument holds in the case when $k_2 \neq k_4$

Now since $i \in P(k_1,k_2)$ and $i \notin P(k_3,k_4)$, either $d(i,k_3) > d(j,k_3)$ or $d(i,k_4) > d(j,k_4)$. This is because if both $d(i,k_3) \leq d(j,k_3)$ and $d(i,k_4) \leq d(j,k_4)$, then $d(k_3,k_4) = d(k_3,i)+d(i,k_4) \leq d(k_3,j)+d(j,k_4)$, implying there is a shorter path between k_3 and k_4 through i rather than j which is a contradiction (since j lies on the shortest path between k_3 and k_4). Note it is also a contradiction even when $k_2 = k_4$ (since j lies on the shortest path between k_3 and k_2 and i does not). Thus, $\neg\left(j \preceq_A^{\delta} i \right)$.

Similarly, since $j \in P(k_3,k_4)$ and $j \notin P(k_1,k_2)$, either $d(j,k_1) > d(i,k_1)$ or $d(j,k_2) > d(i,k_2)$. This is because if both $d(j,k_1) \leq d(i,k_1)$ and $d(j,k_2) \leq d(i,k_2)$, then $d(k_1,k_2) = d(k_1,j)+d(j,k_2) \leq d(k_1,i)+d(i,k_2)$, implying there is a shorter path between k_1 and k_2 through j rather than i which is a contradiction. Note it is also a contradiction even when $k_2 = k_4$. Thus, $\neg\left(i \preceq_A^{\delta} j \right)$.

Thus, given any two vertices $i,j \in V \setminus A$, irrespective of whether they are on the same shortest path or different shortest paths, they cannot spoof each other's location. This implies graph G is $(A,0)$-Geostable, i.e.

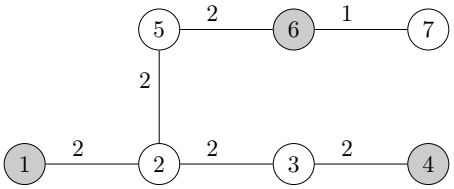
$$\forall_{\substack{i,j\in V\setminus A\\ i\neq j}}\ \exists_{k\in A}\ d(i,k)>d(j,k)$$

However, $V(G):=\bigcup_{k,k'\in A}P(k,k')$ is not a necessary condition for a Geostable graph. $E(G)$ can always contain edge weights / latencies such that a further-off relation is not possible (see next Section for a proof). As a consequence, Theorem 4 is sufficient as it is edge weight agnostic. In Appendix E, we verify Theorem 4 with simulations using real-world data sets and evaluate the spoofing distance as the intersection of A and the geodetic set decreases.

A.1 Theorem 4 is only sufficient for geostability

$V(G):=\bigcup_{k,k'\in A}P(k,k')$ is not a *necessary* condition for a Geostable graph. $E(G)$ can always contain edge weights / latencies such that a further-off relation is not possible.

Proof. See the example below where the shaded vertices depict set A. Note that vertex 7 is not part of any shortest path, but yet vertex 7 is not further off from any other vertex. Similarly, there is no other vertex that is further off from 7.



B Missing Proofs from Section 4

B.1 Proof of Corollary 1

Consider the honest profile $\mathbf{x}^*=(H,H,\cdots,H)$. Note that for any player i, since everybody else is playing honestly under this strategy profile, we have that[7] $p_h(i,\mathbf{x}^*_{-i})=0$. It is trivial now to note that for any player who belongs to a designated region, there is no incentive to deviate. Indeed, such a player is already getting the additional bonus g with probability 1, as there are no false positives under $\mathbf{x}^*$. Therefore, by deviating from honest play, she only increases her probability of losing her deposit, and at the same time, she is not gaining more rewards when not caught. Hence, it remains to look at the incentives of players who do not belong to a designated region. Fix such a player i, and apply now Theorem 1. By using the fact that $p_h(i,\mathbf{x}^*_{-i})=0$, we obtain the desired inequality. □

[7] If all players are honest, then the only way a protocol could mark someone as dishonest could be due to some network error or miscalculations, but we ignore such aspects in this work.

B.2 Proof of Theorem 3

Under the honest profile, the total utility of S is $\sum_{i\in S} u_i(H,H,\ldots,H)$. For each member $i\in S_1$, her utility under honest play equals $(r+g)$ with probability 1. For the members of S_2, it equals r. Therefore, by summing up, we have

$$\sum_{i\in S} u_i(H,H,\ldots,H) = (t-t_s)(r+g)+t_s r \tag{5}$$

If the set S deviates, then we should estimate the expected utility of the entire coalition. For the non-spoofers, which are the members of S_1, they will continue to receive $r+g$. Regarding the set S_2 of the spoofers, there is a Bernoulli trial for each $i\in S_2$, and those who get caught are penalized and lose their deposit d, while the remaining ones receive $r+g$. Therefore, in total the expected utility of the colluding set will be equal to

$$(r+g)\cdot(t-t_s)+\sum_{\ell=0}^{t_s}\binom{t_s}{\ell}\cdot p(t,t_s)^{\ell}(1-p(t,t_s))^{t_s-\ell}[-d\ell+(t_s-\ell)(r+g)]$$

In order for the honest profile to be collusion resistant, we therefore need the last expression to be upper bounded by the expected utility of S under honest play. After carrying out the calculations and some arising simplifications, we obtain the desired inequality. □

C Graph-based versus Coordinate-based

A graph-theoretic model enables us to study the localization problem in a generic sense (for instance, for any network) without restricting us to any specific method of obtaining the RTTs (and thus weights), each of which method has its own set of limitations, as we explain below:

- An analysis using the assumption of proportionality between delays and the 'Great Circle' distance would not generalize to any network. For example, when traveling through the sea, the messages are restricted to submarine cables.
- A speed curve/fit cannot capture the exact network structure. That means a speed curve/fit has to be derived for each specific network, as one single curve cannot be accurate to the same degree in all networks. CBG [14] and [26] even postulate that every node in the network requires a separate delay-distance mapping to achieve a sufficient level of approximation.
- Coordinate-based methods need to solve a trilateration/multilateration problem which can only be approximate. Verifying a location claim means finding the optimum of an over (or under) determined system of constraints, such as in [17].
- In contrast, our theoretical approach requires knowledge of the network structure and network weights a priori. Obtaining reliable ground truth means determining parts of the Internet topology, which can be challenging, especially if high granularity is required.
- In a graph-based approach, participants in the localization protocol need to commit to the finite set of nodes present in the graph, leading to a degree of unavoidable imprecision.

D Overview of Data Sets

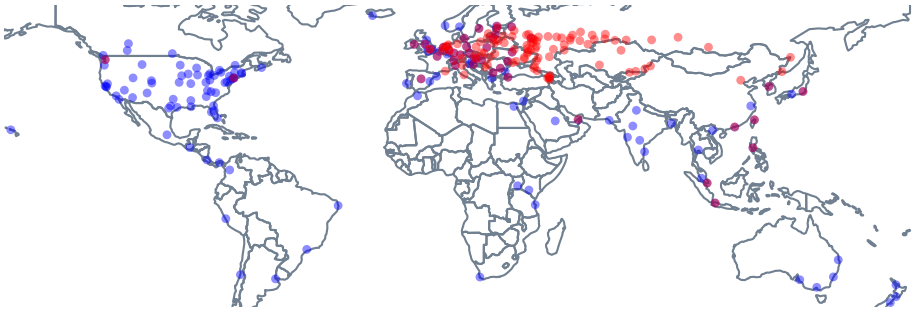


Fig. 4: Overview of the two data sets the simulations are based on. Nodes in data set 1 are depicted as red dots, data set 2 is shown as blue dots.

A graphical overview of the two data sets is shown in Figure 4. Nodes in data set 1 are depicted as red dots, those in data set 2 are shown as blue dots. Data set 1 contains routing information captured as weighted edges between nodes in the graph. Data set 2 includes only an adjacency matrix. The diameter of data set 1 is 257.09ms (Beijing–Seattle, 8684 km). The diameter of data set 2 is 399.09ms (Cape Town–Christchurch, 11007 km).

E Simulations on Geostability and Geodetic Sets

Geostability and Geodetic sets. For data set 1, we study the maximum spoofing distance as a function of the participant set. As shown in Theorem 4, if the localization participants A form a Geodetic set of the graph G, no participant can spoof its location, i.e., the maximum spoofing distance $\delta^{G,A,t,k} = 0$ when $t = 1$ (no collusion) and $k = 1$ (a single mismatched RTT suffices to detect misbehaviour).

We computed the Geodetic set (size=54) for data set 1, and then ran simulations with A sharing its participants from the Geodetic set as illustrated in Figure 5. When the set of player locations in A coincides exactly with the Geodetic set (i.e. overlap fraction = 1.0), the spoofing distance is indeed zero. We also show how the maximum spoofing distance (as a fraction of the diameter of G) changes when A is only an approximate Geodetic set. To do so, we retain a subset of the Geodetic set and fill the remaining positions in A with randomly selected nodes. Our experiments reveal that once the fraction of random locations exceeds 0.5, the maximum spoofing distance increases substantially. Moreover, the effectiveness of geodetic locations diminishes in smaller subsets—when A is a small subset of the Geodetic set, the spoofing distance becomes comparable to that of a fully random selection.

Player Location Distribution. Since players' locations may not form a complete Geodetic set and incomplete Geodetic sets do not maintain low spoofing distances, we

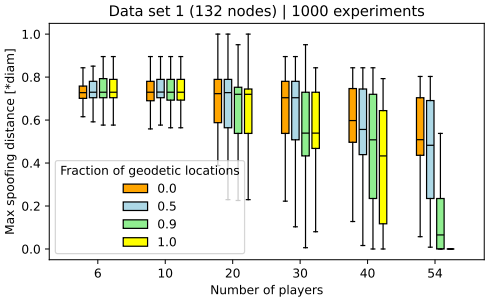


Fig. 5: Geostability for varying number of players at Geodetic set locations, i.e., the fraction defines the number of player locations drawn from the Geodetic set where the remaining locations are i.i.d. The maximum spoofing distance on the y-axis shows $\delta^{G,A,t,k}$ for $t=1,k=1$ and is measured in terms of the diameter of G. A value of 1.0 means that a player can spoof its location accross the entire network.

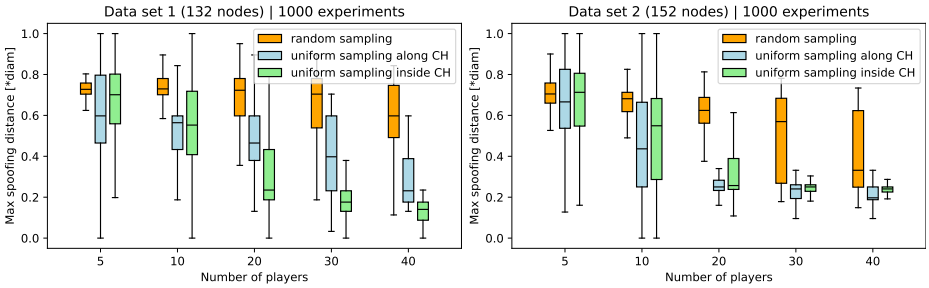


Fig. 6: Maximum spoofing distance ($\delta^{G,A,t=1,k=1}$) depending on the number of players. Player locations are selected using three different sampling methods.

evaluate how the maximum spoofing distance varies under different player location distributions.

Figure 6 again shows $\delta^{G,A,1,1}$ (still in no collusion case) as we vary choose different sampling techniques for A (independent of the Geodetic set). We perform 1,000 simulation runs and report the resulting spoofing distances for varying numbers of players across both data sets. We consider three different sampling strategies for selecting player locations:

1. *Random* — player nodes are sampled without regard to geographic position (orange bars in Figure 6).
2. *Uniform sampling along the convex hull* formed by all nodes (blue bars). This is motivated by the work in [6], which establishes the conditions for secure positioning, and [22], which establishes that source identifiability in Euclidean space is guaranteed if the source lies within the convex hull formed by the observers.
3. *Uniform sampling inside the convex hull area* (green bars). This method draws inspiration from BFT-PoLoc [26] and [21], both of which show improved geoloca-

tion accuracy when challengers are uniformly distributed, increasing the chance that the prover is geographically close to one or more challengers.

Methods 2 and 3 incorporate geographic constraints by limiting the sampling support to either the convex hull's perimeter or its enclosed area. Because the nodes in our data sets are not uniformly distributed and do not naturally align with the convex hull boundaries (see Figure 4), we implement these sampling methods as follows: For method 2, we first compute the convex hull of all nodes and then sample n intermediate coordinates (latitude and longitude) along the hull's perimeter. For each coordinate, we select the geographically closest node, ensuring that no duplicates occur. Method 3 follows the same approach, except that coordinates are sampled uniformly within the convex hull area.

As expected, the spoofing distance generally decreases with more players, since a larger number of players yields more pairwise shortest paths. Additionally, placing players near the "outer edge" of the network (i.e., along the convex hull) increases the likelihood of those paths covering a broad portion of the network. Our simulations confirm this: sampling along the convex hull (method 2) results in shorter spoofing distances than random placement. Sampling within the convex hull area (method 3) tends to produce even shorter spoofing distances in most scenarios (see green bars in Figure 6).

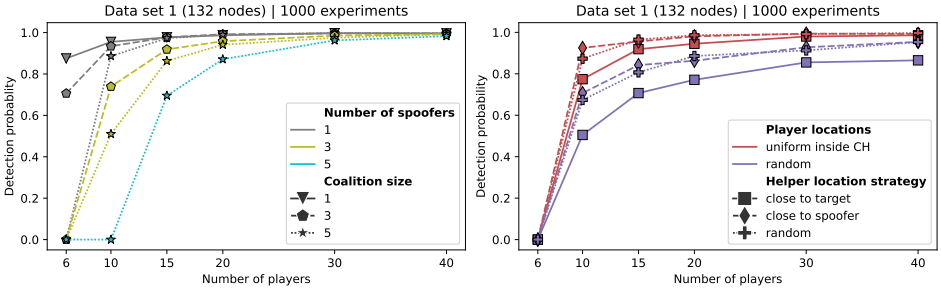
These empirical results show that the difference in spoofing distance between a "good" and a "bad" distribution of player locations can be significant. Assuming that players can strategically position themselves at certain locations prior to participating in the game, a key practical challenge that remains is the efficient discovery or approximation of a suitable set A in a global Internet graph, without incurring excessive measurement overhead or requiring centralized coordination.

F Detection Rates, False Positives and Collusion Strategies Simulated Using Data Set 1

In Section 6, we measured the probability that a dishonest player is correctly identified ($p(t,t_s)$) when attempting to spoof their location, and the probability that an honest player is incorrectly flagged as dishonest (false positive), $p_h(t,t_s)$.

For completeness, we provide the simulation results for data set 1 in Figure 7, which show the average $p(t,t_s)$ when the localization algorithm is configured for low false positives (low $p_h(t,t_s)$). We also set the threshold k to be the number of spoofers plus one, which ensures that spoofers are only flagged when a sufficient number of honest players register inconsistencies.

The results are very similar to those of data set 2 and thus we observe the same trends, i.e., $p(t,t_s)$ increases with the number of players because more players result in more measurement paths, and detection becomes more difficult as the coalition size grows.



(a) Varying number of spoofers and coalition sizes (b) One spoofer in a coalition of size 5 for different helper locations

Fig. 7: Detection probability ($p(t,t_s)$) for $k=t+1$, i.e., no false positives. Player locations are sampled uniformly inside convex hull. Target locations used during spoofing attempts are randomly chosen.

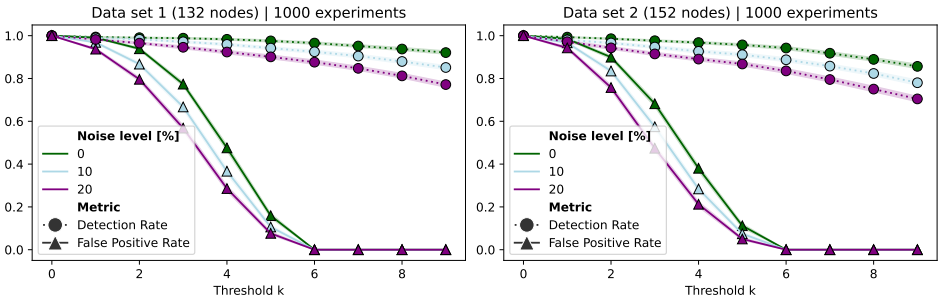


Fig. 8: Detection rate $p(t,t_s)$ (dotted lines) and false positive rate $p_h(t,t_s)$ (solid lines) for varying levels of expected noise. The number of spoofers is 5, the total number of players is 30.

G Network Jitter and Measurement Noise

The localization algorithm (Algorithm 1) relies on inequality conditions to detect mismatches in RTTs. In practice, those inequalities may have to be relaxed to account for variability introduced by routing dynamics and network noise.

We experimentally relax the inequalities in Algorithm 1 by considering noise on the order of $\rho \in [0\%, 10\%, 20\%]$ in the measured round trip times, and therefore we add ρ to the ground truth edge weights e. The inequalities then become

$$rtt_i[j] \leq e[i,j] \cdot (1+\rho) \quad \textbf{or} \quad rtt_j[i] \leq e[i,j] \cdot (1+\rho)$$

In Figure 8, we set the number of players to 30 and the number of spoofers to 5 as an example scenario and show the detection rate and false positives for different noise levels. We observe that accounting for jitter and noise only marginally decreases detection rates and false positives. Overall, a lower false positive rate is generally beneficial, and the effect on detection rate is limited. When adding a margin of 10%, detection rate is lowered by less than 5% (assuming $k=6$, i.e., no false positives).

We note that, in a real system, the effect of noise can be reduced by measuring RTTs repeatedly with multiple ping messages, see, e.g., Verloc [17] where each node sends 200 ICMP requests to any other node and then takes the minimum over the measured RTTs. We did not perform any aggregation of RTTs in our experiments.