Geolocation of Data in the Cloud

Mark Gondree
Naval Postgraduate School
1 University Circle
Monterey, CA 93940
mgondree@nps.edu

Zachary N. J. Peterson
Naval Postgraduate School
1 University Circle
Monterey, CA 93940
znpeters@nps.edu

ABSTRACT

We introduce and analyze a general framework for authentically binding data to a location while providing strong assurances against cloud storage providers that (either accidentally or maliciously) attempt to re-locate cloud data. We then evaluate a preliminary solution in this framework that combines constraint-based host geolocation with proofs of data possession, called *constraint-based data geolocation* (CBDG). We evaluate CBDG using a combination of experiments with PlanetLab and real cloud storage services, demonstrating that we can bind fetched data to the location originally hosting it with high precision. We geolocate data hosted on the majority of our PlanetLab targets to regions no larger than 118,000 km^2, and we geolocate data hosted on Amazon S3 to an area no larger than 12,000 km^2, sufficiently small to identify the state or service region.

Categories and Subject Descriptors

D.4.6 [**Security and Protection**]: Cryptographic controls; E.5 [**Files**]: Backup/Recovery; H.3.2 [**Information Storage and Retrieval**]: Information Storage; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

General Terms

Security, Legal Aspects, Measurement, Experimentation

Keywords

Cloud storage, storage security, data availability, provable data possession, proof of data geolocation

1. INTRODUCTION

Private organizations and governmental agencies with limited storage and IT resources are now outsourcing storage to cloud-based service providers, in an attempt to leverage the manifold benefits associated with cloud services: resource pooling, rapid elasticity, metered service, *etc.* There are legitimate concerns, however, about the implications of using cloud storage services for critical assets. This is especially true in light of recent high-visibility failures, including a massive service outage at Amazon, resulting in the permanent loss of customer data [1, 7]. An Amazon outage in 2008 was due directly to one of its Amazon S3 geographic region centers becoming unreachable for several hours [19].

Increasingly common are cloud service options and service level agreements (SLA) that specify (among other things) the geographic region of a service, at the granularity of a city, state, time zone or political boundary. Geographic region options are provided to help customers achieve a variety of objectives, including performance, continuity and regulatory compliance. For example, a non-U.S. company may want its U.S. customer-serving website located within the continental United States, to improve load time and responsiveness for its target demographic. For contingency planning, a customer may want her data replicated across numerous, geographically distinct locations, for permanence in the face of regional outages or natural disasters. Risk management strategies may be based, in part, on the properties of a specific data center. For example, Amazon GovCloud offers services in a physically separate Amazon service center in Oregon, with specific physical security measures and cleared staff [2, 22]. Further, there are a variety of legal restrictions and protections that may compel a customer to choose to locate data in a specific geographic region. For example, many privacy laws—such as those in Nova Scotia, British Columbia, Australia and soon the EU [16]—require citizens' personal data remain stored within a political border (or, often, that of another nation with comparable protections).

Reliance on a contractual obligation, however, may fail to detect misbehavior (either malicious or accidental) on the part of the service provider. For example, a careless service provider may move client data wholesale, in violation of an SLA, to an overseas data center, to leverage cheaper IT costs. A provider may consolidate data centers or deduplicate client data, undermining those users who are intentionally duplicating data across multiple non-collocated centers for contingency planning. There is concern that the economic incentives for storage providers only align with the goal of preventing reputation-spoiling failures, but do not align with other service characteristics, for which audits, instead, may be appropriate [39]. Indeed, we see these sentiments echoed by potential customers in a variety of sectors. For example, in the United States, the Whitehouse's Federal Cloud Computing Strategy recommends vendors be held ac-

CODASPY'13, February 18–20, 2013, San Antonio, Texas, USA.

countable for service failures, using active SLA compliance monitoring [27]. Likewise, the US Federal Risk and Authorization Management Program (FedRAMP), which establishes a set of government-wide standards for the use of commercial cloud services, mandates the continuous, active monitoring of services [11].

Verifying that a cloud storage service provider is meeting geographic obligations is a challenging problem, and one that has emerged as a critical issue. Benson *et al.* [6] and Peterson *et al.* [35] each, independently, propose using proofs of data possession and host geolocation to bind cloud data to a specific geographic location. Extending this work, we extract a more generic framework for actively monitoring the geographic location of data in the cloud, using latency-based geolocation techniques to implement data geolocation. We contribute the following:

- We relax the adversarial models and assumptions of previous work, and introduce a generalization of the adversarial model of Peterson *et al.* We show that, against a type of strong, covert adversary, the protocol's soundness is limited relative to the block length, the file size, the bandwidth available to the service provider, and the time period across which an audit is performed.
- We present *constraint-based data geolocation* (CBDG), a data geolocation solution that builds on constraint-based techniques for host geolocation. Our methodology is generic enough to use any distance-latency model, including topology-aware models and those that simulate the overhead of specific storage services. As a preliminary approach, we first explore the effectiveness of a simple linear model.
- We validate CBDG using a proof-of-concept simulation, geolocating targets using large messages on the PlanetLab infrastructure. We use these models to successfully geolocate data stored by Amazon Web Services, both with and without observational landmarks collocated with the storage service.

2. SCOPE

Broadly, the problem we consider is how to actively monitor if a storage service provider is meeting its geographic obligations. As with previous work [6, 35], we note that tracking all copies of data in the cloud is not within the scope of our solution, but instead focus on verifying the location of known data copies. The premise of data geolocation is that a provider may have some (economic) incentive for re-locating data in breach of contract, but assuring that all *copies* exist within some geographic region is outside the scope of our work. In particular, data geolocation may be able to detect (in the context of privacy law) non-compliance, *i.e.* that a host serving citizens' data appears to reside outside the borders of its owners' country in violation of a service provider's geographic obligation; it cannot, however, detect compliance, *i.e.* it cannot "discover" copies of the same data that are held, secretly, at some unknown, remote location. We discuss the relationship between our problem and various outstanding legal questions, similar to these, in Section 8.3.

3. BACKGROUND

To implement data geolocation, we combine two previously orthogonal ideas: host geolocation and cryptographic proofs of data possession. Here, we survey previous work in both areas, and highlight those features we leverage in our work.

3.1 Host Geolocation

Discovering the physical location of a host on the Internet is a natural problem for a variety of applications, has been the focus of recent research, and several commercial products now exist providing these services. In a security context, host geolocation has largely been employed to limit online content and services, such as streaming video or online gambling, to specific geographical regions.

Geolocation of hosts on the Internet is currently achieved through a variety of evidence-gathering practices, including mining data from `whois` databases and DNS records, using Internet topology data and through the manual inspection of Internet artifacts (*e.g.* confirming a webpage is written in Chinese). These methods provide a "best guess" based on a small constellation of heuristic evidence, generously assumed to be non-malicious. The only unspoofable, technical method for bounding location on the Internet, however, is active measurement—*i.e.* delay probes from known landmarks—in conjunction with topological information, *e.g.* from path probing and BGP routing views [20, 26, 29, 33].

To build a topology-based solution, Katz-Bassett *et al.* use `traceroute` and latency measurements along a routing path to infer location. Obtaining latencies between intermediate router hops further constrains the possible location of a target. This approach, and subsequent optimizations [44], underlie current state-of-the-art for Internet host geolocation as used in commercial services. The accuracy of these commercial databases, however, is questionable [23, 36]. A study by Siwpersad *et al.* [40] finds that, in querying more than 1.2M IP addresses, approximately 50% of the results from two popular commercial geolocation databases differ by at least 100 km.

Multiple measurements mitigate variable sources of observed delay, *e.g.* congestion, while transmission and processing delay are assumed to be negligible, relative to propagation time. By using multiple landmarks with known positions, delay measurements allow for multilateration of the target's feasible region. The correlation between delay and distance, however, is not always strong due to Internet peering points, topology, and layer-2 traffic engineering [40]. In particular, Internet delays are known to violate the triangle inequality. This is especially true considering the power of an adversarial node against these types of measurement [18].

In this work, we use constraint-based geolocation (CBG) as described by Gueye *et al.* [20]. CBG utilizes delay-based constraints, measured using a set of landmarks at known locations. Real-world network delays rarely correspond to a simple relationship between great-circle distance and speed of light (a physics-based *baseline*), although such baselines produce conservative, upper-bound models. The CBG *bestline* model attempts to be a more realistic relationship between observed network delay and geographic distance.

In the bestline model, all landmarks measure the delay to all other landmarks. Then, each landmark i builds a linear model from its set of latency-distance observations

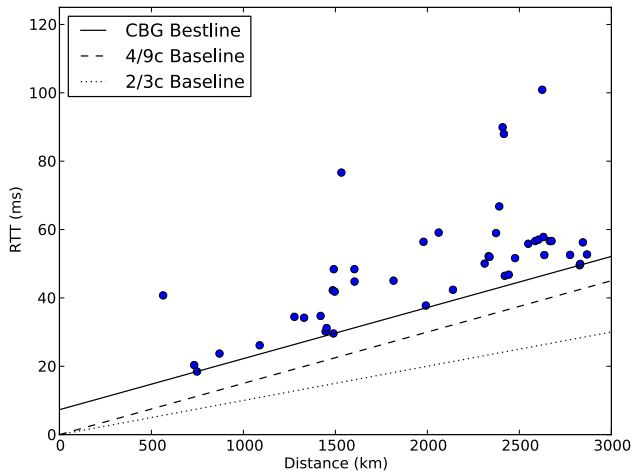


Figure 1: Example scatter plot of distance *vs.* delay from one node (planetlab1.cs.colorado.edu) to 49 others (see Figure 2), with its CBG bestline and two (previously proposed) baselines for Internet delay: $\frac{2}{3}c$ [34] and $\frac{4}{9}c$ [26].

$\{(t_{i,j}, d_{i,j})\}$. This model is characterized by a single line, the CBG bestline (for an example, see Figure 1), defined as the line with non-negative intercept b_i and slope $m_{ij} = (t_{ij} - b_i)/d_{ij}$ closest to that of the baseline slope m,

$$\min_{\substack{b_i \geq 0 \\ m_{ij} \geq m}} \left(\sum_{i \neq j} y - m_{ij}x - b_i \right).$$

Using this model, a landmark can estimate the distance to an unknown target by measuring network delay. Each landmark admits a circular constraint on a target's location, where the circle is centered at the landmark and its radius is equal to the landmark's distance estimation. Define the intersection of these circular constraints to be the target's *feasible region*, a spherical polygon on the surface of the Earth enclosing the target's probable geographical location.

3.2 Provable Data Possession

Provable data possession (PDP) is a class of protocols allowing a client to efficiently audit remote data stores without retrieving the data from the remote server and without requiring the server access the entire file. Recent research has proposed schemes that prove the storage and integrity of remote data [3, 4, 5, 14, 15, 17, 37, 43] and, further, prove the recoverability of the data from incomplete copies [8, 12, 13, 32, 38], the latter termed a proof of retrievability (POR) scheme. Related, proof of ownership schemes [21] allow a server to challenge a client to prove its ownership of a file before allowing storage or access, typically in the context of client-side de-duplication.

All modern PDP schemes follow a similar four-step protocol: (1) the data owner pre-processes the file to create a small, unforgeable *tag*. Clients may now store their data (and in some cases, the tag) remotely, allowing the client to delete its local copy; (2) At a later time, the client (or an auditor, on the client's behalf) issues a *challenge* to the storage server to ascertain the state of the file; (3) The storage server responds to the challenge with a *proof*, which may require the provider to compute some function of the stored file, and; (4) Using the proof and the tag, the client *verifies* if the proof is correct; if so, the proof implies the file is stored at the remote server, intact.

Combining the concepts of PDP with Internet geolocation must be done thoughtfully, providing a new and interesting setting for both problems. Naïvely composing latency-based geolocation with provable data possession, *i.e.* applying each technique serially and independently, provides no assurance. Doing so establishes only two, disconnected facts: first, an unmodified copy of the data exists *somewhere* and, second, the responding server exists within some feasible region. We attain no strong binding between the location and the data. In particular, the geolocated server may be acting as a proxy, relaying the PDP challenges to a server at some alternative location.

4. SECURITY MODEL

In their 2010 paper, Gill *et al.* [18] explore the power of an adversarial host to artificially influence its location as inferred by delay- and topology-based geolocation techniques. They find an adversary can manipulate the perceived location of a target by altering the delay observed by each landmark; however, the adversary may only do so by *adding* delay. Assuming that the attacker does not control the environment during model building, this effectively limits the adversary to causing distance overestimation from observed latencies. Since the constrained region size grows in proportion to the amount of delay added by an attacker, Gill *et al.* observe that using additive delay to relocate a target by 3000 km will result in a median feasible region with 10^7 km^2 area, *i.e.* will inflate the feasible region to roughly the size of Europe. These limitations work in our favor. Given that the goal of our work is to place data *within* some boundary, the ability to grow the feasible region to include points far from the true location serves no useful purpose to our adversary. The work of Gill *et al.* indicates that even a sophisticated geolocation adversary, *i.e.* one who has knowledge and control over the network topology, has no significant advantage in undermining the goals of data geolocation. Alternatively, Gill *et al.* demonstrate that relatively small relocations are difficult to perceive: they demonstrate an attack—where an adversary may move less than 1,000 km without detection—that is successful 74% of the time. Thus, as expected, data geolocation has practical limits when the target must be placed within some small, bounded area or when the target exists near the boundary of a geographic obligation.

Assumptions

For latency-based data geolocation, we make the following assumptions: (1) all data are held, jointly, by some set of target data centers whose physical distance from one another is remote enough to be distinguishable (with a high confidence) by latency-based geolocation; (2) the cloud provider does not have a high-bandwidth out-of-band channel between its centers[1]; (3) the auditor controls a set of semi-trusted, geographically distributed landmarks; and (4) the adversary does not control the entire network environment, *i.e.* the Internet. Assumptions (1-3) are explicit, while (4) is im-

[1]This assumption—that remote sites are not connected by a private network, of significantly better quality than the Internet—is necessary for delay-based IP geolocation (and our work); we acknowledge, however, that providers renting dark fiber may undermine such an assumption.

plicit, in previous data and host geolocation work [6, 18]. For constraint-based geolocation, we add that (5) during an audit, misbehavior (when detected) has a high probability of being detected by all landmarks. In our model, as long as misbehavior has a high probability of being detected by *some* auditor, then assumption (5) can be satisfied by selecting parameters such that every auditor has a high detection probability. In contrast to previous work: we do not require landmarks to be collocated with the targets; we do not need to assume the locations of all data centers are known in advance; we do not need to restrict data movement to other, known targets; we do not need to build a model of the target by interacting with it; we do not require observing the target during a period in which its behavior is presumed honest or otherwise "normal;" and we do not need to use landmarks (collocated or otherwise) running on infrastructure owned by the target provider. Instead, we build a model of the environment using semi-trusted, geographically dispersed, remote landmarks to geolocate data that may have been relocated to arbitrary, new, and previously unknown locations.

Like Benson *et al.* [6] and our previous work [35] we consider an adversary who may deviate from the protocol during an audit. In particular, Benson *et al.* allow the adversary to fetch blocks from a remote location (in violation of the protocol), when a challenge cannot be satisfied using blocks local to the target. (It is not possible, however, for the storage provider to fool an audit by quickly responding with arbitrary data; per-block authentication protect data authenticity.)

Practical Adversaries

Like Benson *et al.*, we consider economically rational adversaries, under the belief that even untrustworthy providers will avoid transmitting or storing data needlessly, *i.e.* purely for the sake of misbehavior. In Section 7, we describe a stronger form of adversary that is able to misbehave without detection if its bandwidth is large (relative to that of the landmark). It is unlikely, however, that this adversary's misbehavior is economically advantageous. In particular, the target may move portions of an archive to geographically remote locations but, during an audit, it must relocate the data back to the target. For an economically rational adversary, this limited misbehavior can be made punitively expensive.

The most conservative approach to auditing using latency-based constraints is to consider the union of all constraints, rather than the intersection, *i.e.* if even one landmark believes a block is being proxied from a remote location, then the feasible region should expand to include that area. Our assumption that misbehavior, when it occurs, is likely to be detected by all landmarks, allows us to consider the intersection of constraints generated by each landmark's audit. With no additional assumptions, each landmark need only consider the maximum latency observed to produce a constraint. Each landmark might assume that if the majority of blocks appear to be local, then its likely all blocks are local; then, it need only consider the median latency to derive a constraint. A stronger landmark assumption is that, if *any* block appears to be local then all blocks are likely to be local; thus, it need only consider the minimum latency observed. For an economically rational adversary, it may not be advantageous to store fractions of a file across diverse locations. Thus, these become reasonable landmark assumptions during constraint generation. More generally, a landmark could generate a constraint corresponding to some α confidence interval, such that all landmarks believe, with α confidence, the derived feasible region contains the target. Alternatively, rather than a strict intersection, one might derive the feasible region that the majority of landmarks believe, with α confidence, contains the target. Gill *et al.* propose the constraint radius itself might be an indicator of confidence, and that a threshold-based criteria may be used to accept or reject constraints when generating the feasible region [18].

5. CBDG

We propose *constraint-based data geolocation* (CBDG), a general protocol that binds latency-based geolocation techniques with a probabilistic proof of data possession. We combine these techniques to develop a protocol with the assurances of each, allowing us to place data geographically within a region while proving them to be authentic. In particular, leveraging CBG allows us to detect when target data changes location, arbitrarily. Further, when target data is replicated in multiple places, using enough geographically dispersed observational landmarks has the potential to locate the target at these different positions (or fail to locate the target at any single geographic position).

General Framework

Our techniques are not strongly bound to any particular distance-latency estimate model. Theoretically, pure delay-based models have the advantage that they will always be an overestimation of distance based on the additive delays incurred from both deterministic and stochastic network delays. They are susceptible only to attacks that misplace targets at further distances. Other models that perform better in practice—such as latency models incorporating route and path data—appear susceptible to underestimation in the face of an adversary who controls a portion of the network (as a large cloud provider might). In our work, we consider the CBG bestline model due to its simplicity, for ease of comparison with previous CBG research, and as it is foundational to more sophisticated geolocation models [18, 26, 44]. The bestline model is intended to be an improvement on a pure speed-of-light model, providing the most conservative linear model resembling the baseline while underestimating all observed data.

Our techniques are not strongly bound to any particular proof of possession scheme, either. Initially, we have selected a simple MAC-based PDP scheme [25, 32] in which a file F is broken into blocks $\{m_i\}$ and tagged:

$$T_{i,m_i} = \mathsf{MAC}_k(\mathtt{name}_F||i||m_i).$$

In this scheme, the blocks $\{m_i\}$ and tags $\{T_{i,m_i}\}$ are stored on the remote server. To challenge the server, the client chooses c random indices and requests the corresponding block/tag pairs. To verify, the client recomputes each tag and compares it with the response; the audit's soundness is a function of c.

Using a MAC-based proof scheme has the advantage of requiring no server-side computation and little client-side storage: for each challenge, the server merely retrieves the response from storage; to verify the proof, the client stores only $O(1)$ state, *i.e.* the cryptographic material k. As no cloud *storage* service provides the ability to perform arbi-

trary computations, *i.e.* to generate complex proofs, using a MAC-based scheme can be immediately implemented given existing cloud infrastructure. The simplicity of a MAC-based PDP scheme, however, comes at a relatively high communication cost. Using a block size of b bytes, at least $c \times b$ bytes must be transferred (at some cost to the client). Some alternative PDP schemes offer $O(1)$ network complexity, but require performing complex server-side cryptographic computations [3, 38]. We comment on the potential complications of using alternative proofs in Section 8.

Protocol Stages

These independent technologies are brought together to create our CBDG protocol:

1. **Model Building:** The landmarks interact, each building a latency-distance estimation model.

2. **Pre-Process and Store:** The data owner splits the file F into blocks, tags each block and stores the block-tag tuple at the target storage service.

3. **Pre-Audit:** The data owner randomly selects c unique challenges and divides them among the landmarks.

4. **Audit:** Each landmark challenges the target. During challenge i, the landmark records the delay t_i associated with the response (m_i, T_{i,m_i}). Using its latency-distance model, the landmark estimates the distance d_i associated with delay t_i. Landmark ℓ uses its set of estimates $\{d_i\}_\ell$ to generate a circular constraint of radius r_ℓ centered at ℓ. Each landmark returns the data $\{(m_i, T_{i,m_i})\}$ and constraint r_ℓ to the data owner.

5. **Verification:** The data owner verifies the proofs $\{(m_i, T_{i,m_i})\}$ received across all landmarks. If all appear valid, the data owner accepts the proof.

6. **Geolocation:** If the proof is accepted, the intersection of all constraints $\{r_\ell\}$ defines the data's feasible region.

The landmark's constraint r_ℓ may be generated from its observations $\{d_i\}_\ell$ in one of various ways, depending on the adversarial model and the desired feasible region accuracy and precision. We discuss some possible choices related to economically rational adversaries in Section 4.

The model building stage is intended to lower-bound the full round-trip time of a challenge and response through the environment. We denote the time to request and receive a single data block as the *data geolocation round-trip time* (DG-RTT). Let DG-RTT$_{\ell,k}$ be the set of DG-RTT observations made by landmark ℓ when contacting landmark k. Landmark ℓ builds a delay-distance model using the min(DG-RTT$_{\ell,k}$) for each k as its data. Assuming the adversary does not control the entire environment during model building, this provides each landmark with data representing the expected minimal latencies between one another.

Protocol *correctness* requires: (a) the verifier accepts all valid provers and (b) the feasible region contains any provers with α confidence. Protocol *soundness* requires: (a) a cheating prover has only negligible advantage in causing the verifier to accept and (b) a prover fetching its data remotely can appear to be storing the data locally, with only limited success. Under the belief that the adversary is economically rational, that c is chosen so that all landmarks detect misbehavior with a high probability, and that the bestline model underestimates future observations: the landmark can generate the constraint r_ℓ using the minimum (or median) of its d_i estimates, assuming the location of one (or most) blocks can place the location of the entire file with α confidence. We discuss the impact on soundness when weakening these assumptions in Section 7.

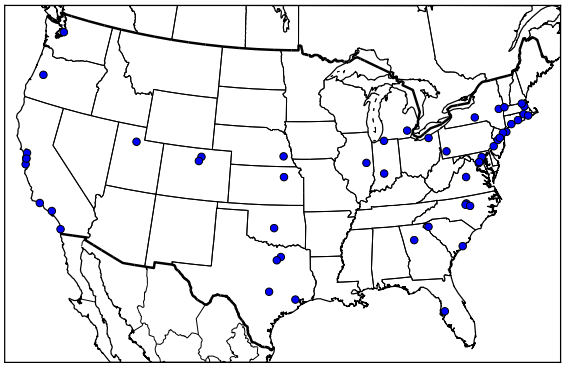


Figure 2: PlanetLab nodes used in experimentation.

We next describe our experiments attempting to validate and assess CBDG. In our analysis, we distinguish between the feasible region's precision and accuracy, as metrics for our protocol's utility. In one sense, if the area encloses the target, then that region is accurate. Gueye *et al.* introduce a different, and heuristic, accuracy metric: the distance of the target from the centroid of the feasible region. Likewise, precision may be related to the area of the feasible region.

6. EXPERIMENTAL RESULTS

To gauge the effectiveness of the proposed techniques, we evaluate them in simulation (on PlanetLab) and in real-world environments (using Amazon S3). We are motivated to discover the simplest, effective data geolocation techniques and our CBDG experimentation follows this motivation. For example, we do not attempt to build a high-fidelity, topology-aware latency-distance model that reflects the provider's overheads: first, that model would likely only be valid in reference to a very weak and specific adversary and, second, the decision to create such a model should be motivated by first exploring much simpler ones. We explore the bestline model, as it is the simplest, non-trivial, conservative delay-based model. We describe the details of our evaluation, next.

Fifty PlanetLab nodes were chosen by hand, based on their geographic diversity and availability (see Figure 2). We choose landmarks in a single country, as we believed this to be a reflection of how CBDG might be deployed: while overseas landmarks may reduce the target's feasible region, particularly for hosts near a border, transferring data internationally during an audit may, in practice, leave it exposed to foreign jurisdictions and subject to divergent (and perhaps conflicting) rules governing protection [31]. While our experimental results are concentrated in the continental United States, we believe our techniques are generalizable to any region where constraint-based geolocation has been demonstrated to be feasible (*e.g.* Western Europe) [20].

On PlanetLab, we utilize a simple TCP challenge-response

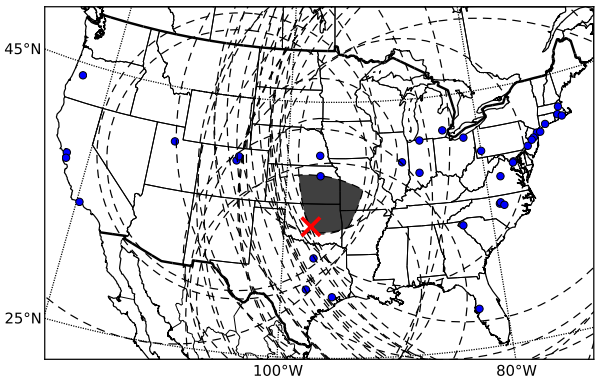


Figure 3: The computed feasible region (grey) for the target data, based our PlanetLab simulation of a CBDG audit, using 32K blocks. The region has area 245,898 km^2 and its centroid is 253 km from the target ($\times$).

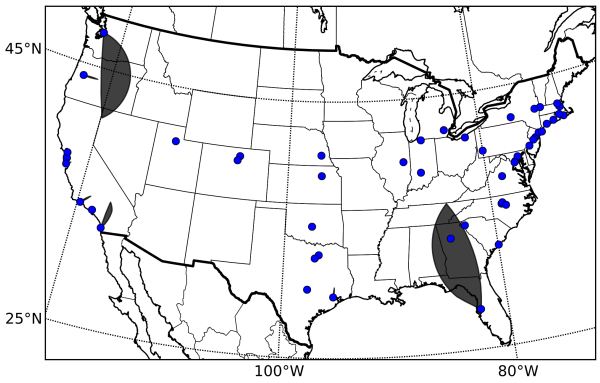


Figure 4: The optimal feasible regions for our selected PlanetLab targets. Note, some regions are so small they are occluded by the target markers.

protocol between landmarks. By using TCP in our measurements, the delay-distance models begin to reflect the protocol overhead associated with interacting with a real cloud storage service provider. As such, our bestline models start to characterize the service behavior of a cloud provider rather than characterize delay artifacts from low-level protocol implementations, such as fragmentation or the maximum transmission unit (artifacts considered by more complicated host geolocation techniques). We do not attempt to model overheads associated with the S3 service or the service infrastructure, *e.g.* delays from IO latency or load balancing.

In each experiment, nodes serve pseudorandom data generated by the `urandom` device. This limits any IO delay that might be incurred by reading on-disk data from the set of PlanetLab nodes. These heterogeneous nodes are known to have diverse performance characteristics, whose IO behaviors may not be representative of a real storage provider's IO delays. We acknowledge this simulation may not be appropriate for those providers whose service can become IO bound or otherwise comes with large delays; consider, for example, the seek times associated with random access using tape storage. We believe, however, that imposing a QoS requirement on the provider may be acceptable in many scenarios, and is reasonable to consider as an initial approach.

Our PlanetLab experiments consider data stored in blocks of size 2^n bytes, where $n = 0, \ldots, 15$, (1 to 32K bytes). In our analysis, we often compare the largest of these (4K, 8K, 16K, 32K) to 64 bytes to compare with previous CBG research. Each pairwise DG-RTT interaction is sampled ten times, choosing nodes in random order, to build the set DG-RTT$_{\ell,k}$. Due to intermittent PlanetLab node failures, not all landmarks participated in all measurements or every experiment. In each experiment, we select one distinguished node to be a target, excluding it from the set of landmarks (*i.e.* during model building), in round-robin fashion.

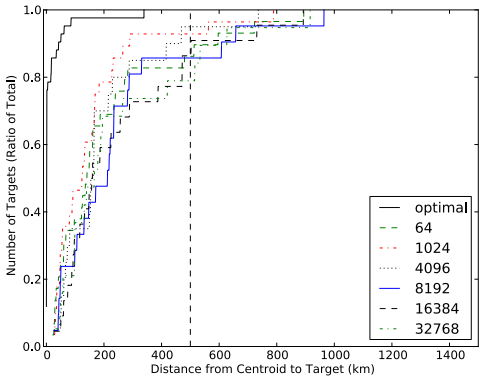
6.1 Simulated Data Geolocation

We investigate the accuracy and precision of target feasible regions in our PlanetLab simulation. For an example feasible region from these experiments, see Figure 3. For each experiment, we consider the feasible region generated when participating landmarks act ideally, generating constraints for the target that perfectly reflect its true distance from the landmark. The resulting intersection is the target's *optimal feasible region*. Due to the geographic interrelationship among landmarks some optimal feasible regions are themselves quite large (see Figure 4). This demonstrates that even in the best case, feasible regions are largely a function of landmark placement. We use this optimal behavior as the primary point of comparison for our experiments.

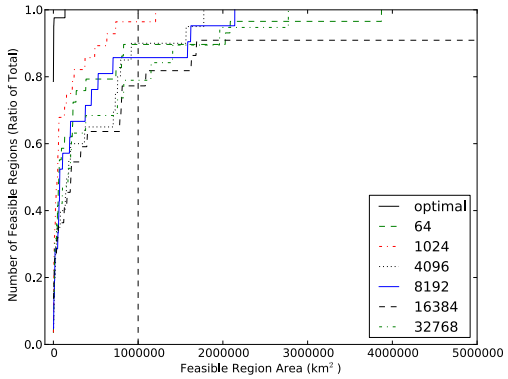
Across all experiments, we see geolocation precision and accuracy that is suboptimal, but relatively similar across block sizes (see Figure 5). In general, accuracy and precision when geolocating with larger block sizes (up to 32K) compare well with that of using CBG-length responses (64 byte blocks). Over 90% of the regions generated during audits on blocks of 32K or smaller have centroids at most 626 km away from their target; over 50% of these regions have centroids no more than 166 km away (see Figure 5(a)). This accuracy is consistent with that of the CBG results of Gueye *et al.*, which claimed an average distance of 78-182 km. The geolocation precision for larger block sizes are similarly favorable. For 32K blocks, 90% of regions are no more than 1,960,510 km^2 in area; 50% are no more than 171,819 km^2 in area (see Figure 5(b)).

6.2 Geolocating Amazon S3

To gauge the usefulness of our techniques in a real-world cloud storage setting, we use our PlanetLab landmarks to geolocate data stored in Amazon's Simple Storage Service (S3) US West-Northern California data center, assumed to be at an unknown location. After interacting with the other landmarks to build a latency-distance model, each PlanetLab landmark fetches a 1GB file in 32K blocks, measuring the DG-RTT of each block request. An HMAC-SHA256 of each block is stored as S3 metadata and returned with each block request. The time to verify the MAC is not considered in the DG-RTT. To ensure our S3 geolocation experiments do not exclusively reflect the utility of collocated landmarks,

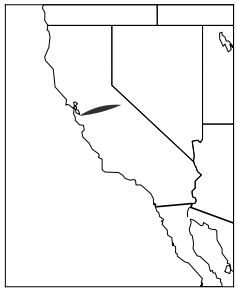


(a) CDF of distance from centroid

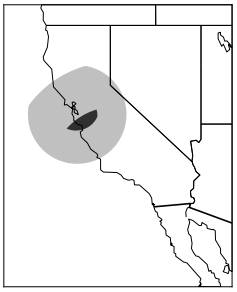


(b) CDF of feasible region area

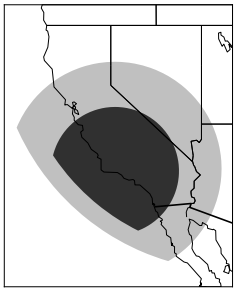
Figure 5: Cumulative distribution functions (CDFs) for feasible region distance and area for 50 PlanetLab nodes over various block sizes.



(a) Optimal without nearby nodes: 5,114 km^2 (5,114 km^2).



(b) With nearby nodes: 11,175 km^2 (243,791 km^2).



(c) Without nearby nodes: 381,607 km^2 (995,967 km^2).

Figure 6: Feasible regions for Amazon's S3 Northern California data center using minimum (dark grey) and median (light grey) DG-RTT and optimal measurements. For each, area of the minimum- (median-) derived region is indicated.

we remove all Northern California PlanetLab landmarks[2] and evaluate, separately, the effects of nearby landmarks.

When geolocating S3 data with nearby nodes, we see very high precision at only a small cost in accuracy, *i.e.* underestimation (see Figure 6(b)). Results compare well with the optimal feasible region (Figure 6(a)). Comparing to the U.S. CBG results of Gueye *et al.* , this S3 feasible region area is smaller than the majority (approximately 65%) of those experimental results [20]. Since Gueye *et al.* found European targets generated, on average, smaller feasible regions, this suggests that CBDG may perform equally well in that setting, if it follows the U.S. data set trend.

Unsurprisingly, nearby landmarks contribute significantly in restraining the feasible regions for CBDG. In particular, we see an order of magnitude improvement by including nearby landmarks. We remark that nearby landmarks are substantially different from collocated landmarks, in terms of our model assumptions. Collocated nodes run on the service provider's infrastructure, co-resident with the storage service. Its unclear what assumptions are implicit, when landmarks themselves are under the control of the service provider.

[2]Excluding nearby landmarks, the nearest landmark (planetlab1.cs.ucla.edu) is just over 500 km from the target.

7. BEYOND RATIONAL ADVERSARIES

In CBDG, we consider an economically rational adversary, based on the belief that the primary motivation for misbehavior is economic. For these, the cost of misbehavior can be made punitive through regular audit, and certain limited forms of attack are unlikely *prima facie*. We note that this type of adversary is different from other rational adversaries considered in cryptography. In particular, covert adversaries may misbehave arbitrarily, as long as the probability for detecting misbehavior is negligible. Peterson *et al.* hypothesize a type of covert attack where the adversary may pre-fetch remote blocks early in the protocol, in anticipation of a future challenge. We expand on this model, as it is potentially applicable to weakening the assumptions of our setting.

We model the protocol as a c-round interactive proof among the target and the landmarks where, each round, some landmark challenges the target. Let r be the number of remote blocks, *i.e.* those held at some location that is geographically distinct from the target's location. Let time t_i be the time elapsed from the start of the protocol to the end of round i. Thus, the DG-RTT for challenge i isΔ $t_i = t_i - t_{i-1}$. Let β be the number of blocks that can be moved from remote to local storage, per unit of time. Without loss of generality, we assume this rate of transfer cannot be improved using data compression: either we inflate β to reflect the speed-up as-

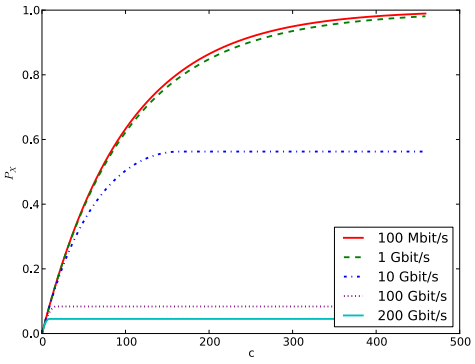


Figure 7: Probability of detecting that no more than 1% of a file is remote before an audit ($r = 1\%$ of n), versus number of challenge rounds (c), for various bandwidths (β); we consider a 1TB file in 4KB blocks, where each challenge round is 50ms.

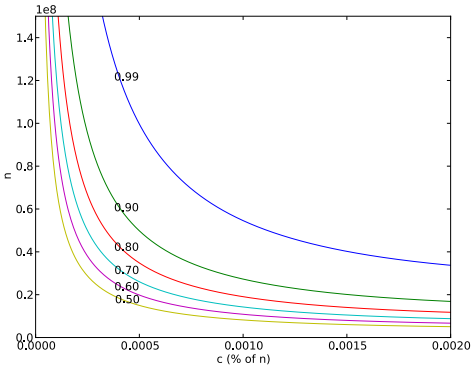


Figure 8: Probability of detecting server misbehavior (P_X) as a function of the number of blocks (n) and the number of challenge rounds (c), for $r = 1\%$ of n; we consider β as a blocks-per-round rate equivalent to the 100Mbit/s bandwidth from Figure 7.

sociated with compressed blocks, or we prevent compression by, say, encrypting all blocks.

Let X be a discrete random variable representing the number of challenged blocks not held local to the target; it is for these blocks that the target will either (i) provide a counterfeit proof that (with a high probability) will be rejected as invalid, or (ii) fetch the data from some geographically remote storage, causing some measurable delay$\Delta\ t_i$ during the round. We compute $P_X = P\{X \geq 1\}$, the probability that at least one of the challenged blocks matches a block that is exclusively held at a remote location.

$$\begin{aligned} P_X &= P\{X \geq 1\} = 1 - P\{X = 0\} \\ &= 1 - \frac{n-r}{n} \times \frac{n-(r-\beta t_1)-1}{n-1} \times \cdots \\ &= 1 - \prod_{i=0}^{c-1} \frac{n - \delta(r-\beta t_i) - i}{n-i} \\ &\text{where} \quad \delta(x) = \begin{cases} x & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

In this model, the time period prior to challenge i may be used to move at most βt_i of the r remote blocks to local storage. Unlike a traditional provable data possession adversary, this model allows the target to "undelete" the blocks that have been locally deleted, at some cost. Letting $\beta = 0$ gives us the soundness of traditional PDP [3]. For $\beta > 0$, the protocol loses the property where querying produces arbitrary soundness amplification. In particular, at the point in the protocol where $\beta t_i > r$, all further queries provide no increase to soundness.

Consider the scenario in which, for any round i, we have $t_i = 1$, *i.e.* every takes the same "unit of time," during which β blocks can be moved from remote to local storage. This is a scenario where blocks fetched locally from the target yield latencies that are identical, or similar. In this scenario, we have

$$P_X = 1 - \prod_{i=0}^{c-1} p(i) \quad \text{where} \quad p(i) = \frac{n - \delta(r - i\beta) - i}{n-i}.$$

For $i \leq r/\beta$ its the case that $p(i-1) \leq p(i)$, and for $i > r/\beta$, we have $p(i) = 1$. Thus, letting $c_0 = \min(c, 1 + \lfloor r/\beta \rfloor)$, the probability of detecting misbehavior becomes bound by

$$1 - p(0)^{c_0} \leq P_X \leq 1 - p(c_0 - 1)^{c_0}.$$

This behavior can be seen in Figure 7, where the protocol's soundness reaches its maximum at some round c_0 and then becomes constant. We use our bounded formulae to show P_X as a function of n and c in Figure 8. The probability of undetected misbehavior, essentially, is characterized as an interaction between the scale of misbehavior, the relative bandwidth available to auditors and providers, and the duration of the audit.

We note that soundness for misbehavior detection degrades as rapidly as the adversary becomes compliant with the geolocation agreement. Thus, auditing may be seen as enforcing correct behavior, rather than detecting slight misbehavior. Our model may be further generalized to allow parallel challenges from landmarks in each round, thus slowing the degradation of soundness across rounds to provide stronger assurance of detection. We do not, however, explore the parallel challenge strategy here. First, such a strategy requires heavyweight assumptions (strict synchronization between landmarks during the protocol). Second, we are interested in worst-case soundness, so we consider the model that most greatly advantages the adversary.

8. DISCUSSION & FUTURE WORK

8.1 CBG Enhancements

Our CBDG framework is relatively general, and may be adapted to incorporate new constraint-based geolocation techniques. Thus, research into CBG improvements and new delay-distance estimation techniques may immediately benefit the problem of data geolocation. We highlight several avenues for constraint-based geolocation research, based on our experiences with CBDG.

In our PlanetLab-based evaluation, we choose a rather arbitrary, convenient set of landmarks in the continental United States. It became clear, however, that landmark selection is an enormous factor affecting the precision of constraint-based geolocation. The reasons for this are many: node responsiveness, quality of network connection, physical location relative to adjacent nodes, *etc.* Even perfectly

predicting landmarks that are geographically distant from the target will generate huge feasible regions: a single East Coast landmark auditing a West Coast target should produce a feasible region encompassing most of North America and parts of Greenland.

We found, in practice, that some landmarks were much more valuable than others. Let a landmark's *influence* be defined as the percentage by which the feasible region's area is reduced when the landmark's constraint is included in the total set of constraints. For those landmarks participating in an experiment, let the optimal feasible region be the region constrained if all participating landmarks provided ideal distance estimates. In this ideal simulation, we find some landmarks—due only to their participation and geographic location—are sometimes very influential. Some, alone, constrain the region by more than 80% (see Figure 9(a)). On average, as might be expected, no single landmark appears strongly influencing when all landmarks behave ideally. In comparison, we see numerous strongly influencing landmarks in our real experiments, including landmarks that (when they influence at all) are strongly influencing on average (see Figure 9(b)). Not only are the constraints from individual landmarks strongly influencing, we find the bestline models themselves are strongly influenced by single landmarks during model building.

We feel that landmark placement for geolocation warrants independent study, with particular attention to both *coverage* (how to place landmarks to geolocate arbitrary targets within some boundary, with high precision) and *robustness* (how to place landmarks so geolocation is precise, despite some of the landmarks failing or under-performing). Related "placement" and visibility problems, *e.g.* the art gallery problem, do not seem to directly apply to our setting.

We feel that constraint-based geolocation might also benefit from improved distance-latency estimate models. The bestline method, in particular, largely ignores a wealth of latency data, and becomes entirely defined by those small set of data points able to bound a line that, by definition, resembles the speed-of-light baseline model. Alternative models might build and utilize sets of linear models, each of which is appropriate for distance predictions on different ranges of latencies: for ranges on which the model has very little data, it might use a very conservative linear estimate; for ranges on which the model has many samples, it might contribute a more aggressive constraint estimate.

8.2 Alternative Proofs of Possession

One drawback of CBDG is the relatively high network overhead required to perform an audit, especially in comparison with recent proof of possession schemes capable of $O(1)$ network communication [3, 38]. In our scheme, the high network complexity is strategic, to relax the computational burden on the target: the server performs only computations related to fetching data blocks. Of course, many web service companies providing cloud storage also provide computation services. For example, Amazon Web Services offers both the Simple Storage Service (S3) and the Elastic Compute Cloud (EC2). Thus, we might develop a collocated PDP service, leveraging more complex data possession proofs in CBDG.

It may be possible to augment latency-based geolocation models with fine-grained measurements of server-based computation delay, to isolate and identify those components of latency associated with cloud storage infrastructure and those induced by server-side cryptographic computations. Such a modeling strategy would necessarily need to focus on tight lower-bounds, since permitting variable latencies from the target or lengthening the audit window will open opportunities for misbehavior from very strong adversaries. This type of latency decomposition has been explored by the storage and network communities, in similar contexts. For example, delay-based techniques have been recently employed to determine if data has been duplicated within a remote data center [9]. We believe such fine-grained decomposition of delays may allow CBDG protocols to utilize more complex proofs of possession schemes—such as those compressing their responses using homomorphic signatures, those with multiple simultaneous challenges, *etc.* For example, PDP audits that measure above a certain (pre-computed) threshold of delay may suggest the target is fetching data from a remote location.

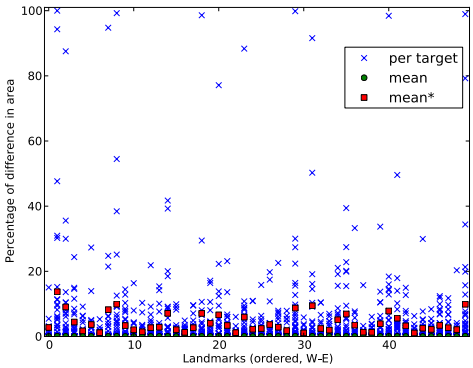
It may also be possible to develop proofs of possession leveraging new assumptions. For example, proofs might constructively leverage un-clonable, tamperproof devices operating on-site at the storage service provider, binding *computation*, rather than data, to a location. Consider a trusted, un-clonable physical device with a GPS receiver at the remote storage server: clients may challenge the server to perform operations on their data using this device, providing an authentic proof of location, albeit under a new and different security model (*e.g.* civilian GPS signals may be spoofed or delayed).

8.3 Data Geolocation in a Legal Context

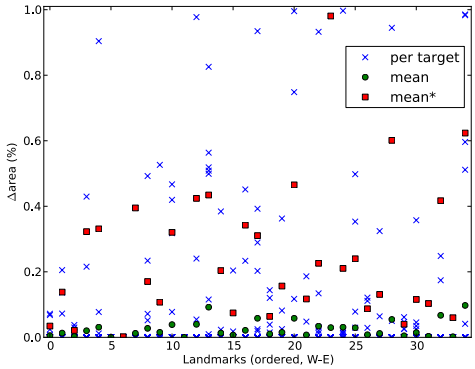
We previously proposed the concept of *data sovereignty*, with some initial ideas on how it can be attained using data geolocation [35, 24]. In a legal context, the traditional notion of sovereignty is often defined by two rights: a positive right, which allows a legitimate authority exclusive claim to an object, and a negative right, which states that no other authority may lay claim to that object. Legal tests of sovereignty may be applied naturally to tangible objects, such as property or chattel. Applying these historic notions of sovereignty to data stored in the cloud, however, fail both legally and technically, calling into question the appropriateness of the term in the context of this and similar work.

While laws that establish exclusive, positive ownership rights—founded in intellectual property law, data protection law and confidentially law—grant data owners rights similar to that of physical property, we are unable to enjoy traditional notions of sovereignty due to ambiguities in cloud data's jurisdiction, making an owner unable to exclude another party's interest. Consider the US PATRIOT Act, which has been recently used to justify the seizure of data, anywhere in the world, if the data center is owned and operated by a US-based company, irrespective of the data's owner [28]. Further, due to an aging Electronic Communications Privacy Act, data residing in the cloud are currently treated jurisdictionally different than data on privately owned devices. Interestingly, only when the owner of data is a government do historic notions of sovereignty begin to make sense.

Similar limitations arise when using technology to assert positive and negative rights on data stored remotely. While our data geolocation protocol makes a small step towards establishing positive rights (perhaps, through some limited theory of jurisdiction), we know of no technology able to



(a) Influence of landmark on optimal feasible regions.



(b) Influence of landmark on measured feasible region.

Figure 9: For each of our PlanetLab landmarks, we plot how much it influenced the actual measured and theoretically optimal feasible regions in our experiments over 4K blocks (Section 6.1). For each, we show the landmark's mean influence and the mean of its non-zero influence (mean*).

track and control the flow of all copies of data in an open system, such as an Internet-based cloud. Indeed, imposing such a technology may be in conflict with the economic benefits of existing cloud models. Cloud providers optimize by leveraging dynamic resource allocation across nodes, moving data and computation to where it is cheapest.

As described in the Introduction, the inertia for organizations to move to cloud-based storage in increasing, but many cloud computing strategies are complicated by a lack of a internationally recognized understanding of data jurisdiction. If these issues are not solved, both legally and technically, cloud storage service providers may be relegated to offering domestic-only services, limiting appeal and increasing costs.

We posit one way forward may be an intergovernmental statement of recognition pertaining to cloud services, clarifying which entities have an exclusive jurisdiction over data in transit, storage, and processing. It will be necessary to avoid multiple jurisdictions. To do so, may require readjustments to law enforcement and anti-terrorism efforts (to define minimum requirements for surveillance and seizure of cloud infrastructure), building mutual trust through uniform due process. It is unrealistic to assume that any common understanding of data jurisdiction will be established irrespective of data's physical location. We believe our contributions, and data geolocation in general, provide an important first step in developing methods for establishing the data-location binding, which may be leveraged by future policy. For further reading on the subject, the authors recommend Irion's recent paper on the subject [24].

9. RELATED WORK

Bowers *et al.* propose an approach to verify a cloud storage service provider is replicating data across multiple drives through fine-grained measurements of delay [9]. Their technique, called *Remote Assessment of Fault Tolerance* (RAFT), yields a tool intended to break the common abstraction of the cloud, through an interactive challenge and response protocol, much like CBDG. The idea of having a simple cloud abstraction when we want it, and removing it when we don't, is a powerful one.

The use of semi-trusted landmarks has been at the foundation of many of geolocation solutions [33, 20, 44, 29]. In wireless networks, hidden landmarks (or, "hidden, mobile base stations") have been used in a slightly different model of geolocation that is robust against a small number of colluding adversaries [42]. Recent results in *position-based cryptography* present some interesting positive results in the "bounded storage model" that are secure, even when considering a very strong type of adversary—capable of breaking nearly all previous geolocation strategies—that is able to clone itself at multiple, specific, hidden locations [10]. Interestingly, this adversary does not necessarily undermine CBDG's goal, which is merely to determine that the adversary is inside some bounding area (not to determine its specific position in this area); therefore, we believe our guarantees may be achieved in a weaker model.

Tools to actively monitor real cloud performance or SLA compliance—such as *CloudCmp* [30], *SLAm* [41] or any of various commercial monitoring services—do not yet offer support for checking compliance with respect to data durability or location clauses of an SLA. Most tools do monitor certain QoS metrics potentially relevant to inferring geolocation and data presence, such as up-time and end-to-end response times. Thus, extending support to monitor data geolocation is quite natural. Established commercial SLA monitoring services provide natural partners for outsourcing data audits or for acting as semi-trusted landmarks capable of participating in data geolocation protocols.

10. CONCLUSION

We have proposed and investigated a method for binding data in the cloud to a location, admitting strong assurance to both data integrity and location. Our initial approach using constraint-based geolocation with proofs of data possession appears promising. We attempt to weaken assumptions in previous data geolocation work—*i.e.* those related to collocating landmarks at the target or running observational nodes on adversarial infrastructure—as we believe this to provide stronger assurances, possibly at the cost of precision. We are particularly interested in protocols with higher accuracy and assurances against stronger adversaries. Toward this, combining CBDG with the replica management techniques of Benson *et al.* or more advanced geolocation techniques seems promising. In particular, one might lever-

age collocated landmarks to build a model of service overhead to more accurately simulate landmark-target interactions during model building; our preliminary results using CBDG using TCP-based models are favorable.

11. ACKNOWLEDGEMENTS

The authors would like to thank Andreas Terzis for his assistance with PlanetLab access and Rob Beverly for early discussion about host and data geolocation. Partial support for this work was provided by the National Science Foundation under award No. 1143573.

12. REFERENCES

[1] Amazon Web Services. Summary of the Amazon EC2 and Amazon RDS service disruption in the US east region. Available at `http://aws.amazon.com/message/65648/`.

[2] Amazon Web Services. Overview of security processes, May 2011. Available at `http://aws.amazon.com/security`.

[3] G. Ateniese, R. Burns, R. Curtmola, J. H. amd Lea Kissner, Z. Peterson, and D. Song. Provable data possession at untrusted stores. In *Proceedings of the ACM Conference on Computer and Communications Security*, 2007.

[4] G. Ateniese, S. Kamara, and J. Katz. Proofs of storage from homomorphic identification protocols. In *Proceedings of ASIACRYPT*, 2009.

[5] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik. Scalable and efficient provable data possession. In *Proceedings of the International Conference on Security and Privacy in Communication Networks*, 2008.

[6] K. Benson, R. Dowsley, and H. Shacham. Do you know where your cloud files are? In *Proceedings of the ACM Cloud Computing Security Workshop*, 2011.

[7] H. Blodget. Amazon's cloud crash disaster permanently destroyed many customers' data. *Business Insider*, April 4 2011. `http://www.businessinsider.com/amazon-lost-data-2011-4`.

[8] K. D. Bowers, A. Juels, and A. Oprea. Proofs of retrievability: Theory and implementation. In *Proceedings of the ACM Workshop on Cloud Computing Security*, 2009.

[9] K. D. Bowers, M. van Dijk, A. Juels, A. Oprea, and R. L. Rivest. How to tell if your cloud files are vulnerable to drive crashes. In *Proceedings of the ACM Conference on Computer and Communications Security*, 2011.

[10] N. Chandran, V. Goyal, and R. M. R. Ostrovsky. Position based cryptography. In *Proceedings of the International Cryptology Conference*, 2009.

[11] CIO Council. Proposed security assessment & authorization for US government cloud computing, November 2010.

[12] R. Curtmola, O. Khan, and R. Burns. Robust remtoe data checking. In *Proceedings of the ACM International Workshop on Storage Security and Survivability*, 2008.

[13] R. Curtmola, O. Khan, R. Burns, and G. Ateniese. MR-PDP: Multiple-replica provable data possession. In *Proceedings of the International Conference on Distributed Computing Systems*, 2008.

[14] Y. Deswarte, J.-J. Quisquater, and A. Saïdane. Remote integrity checking: How to trust files stored on untrusted servers. In *Proceedings of the Conference on Integrity and Internal Control in Information Systems*, 2003.

[15] C. C. Erway, A. Küpcü, C. Papamanthou, and R. Tamassia. Dynamic provable data possession. In *Proceedings of the ACM Conference on Computer and Communication Security*, 2009.

[16] European Commission. Regulation of the european parliament and of the council on the protection of individuals with regard to the processing of personal data and on the free movement of such data (general data protection regulation). Directive 95/46/EC, 2012.

[17] D. L. G. Filho and P. S. L. M. Barreto. Demonstrating data possession and uncheatable data transfer. Cryptology ePrint Archive, Report 2006/150, 2006.

[18] P. Gill, Y. Ganjali, B. Wong, and D. Lie. Dude, where's that IP? Circumventing measurement-based IP geolocation. In *Proceedings of the USENIX Security Symposium*, 2010.

[19] N. Gohring. Amazon's S3 down for several hours. *PC World*, Feb 15 2008. `http://www.pcworld.com/businesscenter/article/142549/amazons_s3_down_for_several_hours.html`.

[20] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida. Constraint-based geolocation of Internet hosts. *Transactions on Networking*, 14(6), December 2006.

[21] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg. Proofs of ownership in remote storage systems. In *Proceedings of the ACM Conference on Computer and Communications Security*, 2011.

[22] D. Harris. Amazon targets US government with GovCloud. *The New York Times*, August 2011. `http://nyti.ms/y6A0ZH`.

[23] B. Huffaker, M. Fomenkov, and kc claffy. Geocompare: a comparison of public and commercial geolocation databases. In *Proceedings of the Network Mapping and Measurement Conference (NMC)*, 2011.

[24] K. Irion. Government cloud computing and the policies of data sovereignty, 2011. Available at `http://ssrn.com/abstract=1935859`.

[25] A. Juels and B. S. Kaliski Jr. PORs: Proofs of retrievability for large files. In *Proceedings of the ACM Conference on Computer and Communications Security*, 2007.

[26] E. Katz-Bassett, J. P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and Y. Chawathe. Towards IP geolocation using delay and topology measurements. In *Proceedings of the Conference on Internet Measurement*, 2006.

[27] V. Kundra. Federal cloud computing strategy, February 2011. Available at `http://www.cio.gov/documents/federal-cloud-computing-strategy.pdf`.

[28] K. E. Kushida, J. Murray, and J. Zysman. Diffusing the cloud: Cloud computing and implications for public policy. *Journal of Industry, Competition and Trade*, 11(3), 2011.

[29] S. Laki, P. Matray, P. Haga, I. Csabai, and G. Vattay. A detailed path-latency model for router geolocation. In *Proceedings of the International Conference on Testbeds and Research Infrastructures for the Development of Networks Communities and Workshops*, 2009.

[30] A. Li, X. Yang, S. Kandula, and M. Zhang. CloudCmp: Comparing public cloud providers. In *Proceedings of the Internet Modeling Conference*, 2010.

[31] Microsoft Corporation. Building confidence in the cloud: A proposal for industry and government action to advance cloud computing. Technical report, Microsoft Corporation, January 2010.

[32] M. Naor and G. N. Rothblum. The complexity of online memory checking. *Journal of the ACM*, 56(1), 2009.

[33] V. N. Padmanabhan and L. Subramanian. An investigation of geographic mapping techniques for Internet hosts. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2001.

[34] R. Percacci and A. Vespignani. Scale-free behavior of the internet global performance. *European Physical Journal B*, 32(4), 2003.

[35] Z. N. J. Peterson, M. Gondree, and R. Beverly. A position paper on data sovereignty: The importance of geolocating data in the cloud. In *Proceedings of the USENIX Workshop on Hot Topics in Cloud Computing*, 2010.

[36] I. Poese, S. Uhlig, M. A. Kâafar, B. Donnet, and B. Gueye. Ip geolocation databases: unreliable? *Computer Communication Review*, 41(2), 2011.

[37] T. Schwarz, S.J. and E. L. Miller. Store, forget, and check: Using algebraic signatures to check remotely administered storage. In *Proceedings of the IEEE International Conference on Distributed Computing Systems*, 2006.

[38] H. Shacham and B. Waters. Compact proofs of retrievability. In *Proceedings of ASIACRYPT*, 2008.

[39] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan. Auditing to keep online storage services honest. In *Proceedings of the USENIX workshop on Hot Topics in Operating Systems*, 2007.

[40] S. Siwpersad, B. Gueye, and S. Uhlig. Assessing the geographic resolution of exhaustive tabulation for geolocating internet hosts. In *Passive and Active Network Measurement*, 2008.

[41] J. Sommers, P. Barford, N. Duffield, and A. Ron. Multiobjective monitoring for SLA compliance. *Transaction on Networking*, 18(2), 2010.

[42] S. Čapkun, M. Čagalj, and M. Srivastava. Secure localization with hidden and mobile base stations. In *Proceedings of the IEEE International Conference on Computer Communications*, 2006.

[43] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou. Enabling public verifiability and data dynamics for storage security in cloud computing. In *Proceedings of the European Symposium on Research in Computer Security*, 2009.

[44] B. Wong, I. Stoyanov, and E. G. Sirer. Octant: A comprehensive framework for the geolocalization of internet hosts. In *Proceedings of the USENIX Networked Systems Design and Implementation*, 2007.