Irain *et al. Journal of Cloud Computing: Advances, Systems and Applications* (2017) 6:31
DOI 10.1186/s13677-017-0095-y

Journal of Cloud Computing:
Advances, Systems and Applications

REVIEW Open Access

Landmark-based data location verification in the cloud: review of approaches and challenges

Malik Irain* , Jacques Jorda and Zoubir Mammeri

Abstract

Data storage on the cloud is growing every day and many companies, administrations, and individuals are now outsourcing storage of their data to large-scale Cloud Service Providers (CSP). However, because of today's cloud infrastructure virtualization, data owners cannot easily know the location where their data are stored. Even in case of the establishment of a strong Service-Level Agreement, which includes an initial guarantee regarding data location, the CSP may then move data to another location, like another country, in order to cut storage costs or for any other reasons, including backup mistakes and fraudulent use of data. Data location verification is required due to legal, privacy, and performance constraints. Recently "Where are my data located in the cloud?" has become a challenge and solutions have been proposed to verify data location, under given assumptions regarding CSP behavior. The objective of this paper is twofold: propose a comprehensive classification of the location verification approaches and discuss their vulnerabilities regarding malicious CSP attacks. Location verification solutions may be Framework-based, Hardware-based or Landmark-based. This paper addresses only landmark-based approaches for their deployment flexibility and low cost.

Keywords: Cloud computing, Data location verification, Landmark-based approaches, Cloud storage, Security attacks

Introduction

Nowadays, cloud computing has become one of the pillars for our computer-based society. The cloud provides shared processing and data storage resources to computers and other IoT devices connected to Internet. Broadly, cloud computing can be categorized into three classes depending on abstraction level:

- Infrastructure-as-a-Service (IaaS), where Cloud Service Provider (CSP) offers "raw materials" such as virtual machines (CPU cores, RAM) or virtual storage. Well known IaaS are Amazon Elastic Cloud Compute [1] and Google Compute Engine [2].
- Platform-as-a-Service (PaaS), where CSP provides a platform allowing users to manage, create and run applications without managing the associated infrastructure. Well known PaaS are Heroku [3] and Microsoft Azure [4].
- Software-as-a-Service (SaaS), which means that CSP provides applications (or services) to users. Well known SaaS are Trello [5] and Google Suite [6].

Cloud usage is growing every day and many companies, administrations, and individuals (all of them are simply called users in the sequel) are now outsourcing storage of their data to large-scale distributed storage systems. Such users are thus relieved of tasks related to the management and maintenance of underlying storage equipment. The counterpart is that they lose some control on their data and they have to trust CSPs, thus resulting in some users remaining reluctant to cloud usage. Consequently, for the cloud to be more widely accepted, users can enforce their requirements through QoS clauses including data location.

Why data location should be known to data owners?

Data collecting, storage, and processing are of prime importance in modern societies, governments, and individuals. Holding data is power whatever is the field

*Correspondence: malik.irain@irit.fr
Toulouse Institute of Computer Science Research, University of Toulouse, Toulouse, France

(health, science, technology, national security, military, ...). Laws applicable to data are generally the ones of CSP's country, which may be different or worse in contradiction with the ones in data owners' country. In addition, because of today's cloud infrastructure virtualization, data owners cannot easily know locations of their data. That is why countries worldwide require compliance with their specific laws regarding data storage and processing.

Objectives of data protection laws [7–9] are mainly:

- Data sovereignty: governments must provide and enforce laws to guarantee data independence to their citizens and companies in order to avoid particularly data access deny by foreign authorities. Indeed, in case of conflicts between data owners' country and CSP's country, the latter may totally or partially deny data access.
- Data protection: any government must protect data in case of war or any critical conditions, i.e. governments must protect citizens and societies as well as their data and do not rely on other countries to provide the required protection.
- Privacy protection: data belonging to citizens and strategic companies must be protected against accesses from abroad. For example, national companies should not store their data in other countries where local governments could redistribute stored data to local competitors.

In addition to data protection laws, data location may be required in order to consider performance issues [10]: the closer to users are the data, the faster is the access. Also, when data are stored in the same country, it is easier to find alternate routes if some routers fail, whereas between countries Quality-of-service is easily affected in the event of links and routers failures.

Why data location verification is needed?

When a contract is agreed between data owner and CSP, the latter is assumed to comply with contract rules. For example, the following is posted on AWS's webpage: "Customers choose the region(s) in which their customer content will be stored. We will not move or replicate customer content outside of the customer's chosen region(s), except as legally required and as necessary to maintain the AWS services and provide them to our customers and their end users" [11].

However, even in case of the establishment of an SLA, which includes an initial guarantee requirements regarding data location, CSP may move data to another location, like another country, in order to cut costs, by mistake or for malicious reason. For example, in November 2016, Russia's communications regulator ordered public access to LinkedIn's website to be blocked to comply with a court ruling that found the social networking firm guilty of violating a data storage law, which stipulates the personal data of Russian users must be stored on the territory of the Russian Federation [12]. Consequently, data users should deploy mechanisms to verify, at any time, that their CSP is complying with data location requirements and not blindly trust it.

SLA and requirements on data location verification

Compliance with data protection laws requires first the establishment of an SLA (Service Level Agreement) between data owners and CSPs. Such SLAs should include a **data location clause**, which clearly specifies **geographic location(s) where personal data may be stored** [13].

Geographic locations may be specified in various ways using GPS coordinates or country, city, and county names. For example, as of December 2016, AWS (Amazon Web Services) enables its clients to specify their data storage region according to a location list: for Europe region, datacenters in Ireland, Frankfurt, and London may be chosen, and for US, datacenters on East and West costs may be chosen.

In addition to data location(s), SLA includes a list of IP addresses or domain names, which enable users access their data—not to verify locations. Notice that an SLA may include an IP address of a proxy located in a geographic region, which differs from the one of physical data server. When commercial CSPs are of concern, proxy IP addresses do not provide (sufficient) information to locate CSP's data servers. Indeed, because of system virtualization and physical infrastructure protection, the IP addresses of physical data servers are not public. Consequently, location verification approaches targeting commercial CSPs should not rely on CSP's IP addresses and domain names included in the SLA to verify data location.

It is worth noticing location verification approaches differ from IP-address-based geolocation (IABG) approaches in their objectives and in the way they deploy landmarks to collect measurements. Location verification aims to check that data are stored in authorized geographic zone—a country, a state, or a county—and not to check a specific data server location, while IAGB aims to find the location associated with a given IP address. Interested readers may refer to CAIDA website, which provides a comprehensive—and nearly exhaustive—review of IABG approaches [14].

Data location verification categories

Recently "Where are my data located in the cloud?" has become a challenge and some authors proposed approaches allowing users to verify data location under given assumptions regarding the CSP behavior,

connection links between users and the CSP, and so on. Those solutions and associated assumptions are investigated in the sequel. Three location verification approaches classes are commonly distinguished (Fig. 1):

- Cloud Framework-based approaches [15–17] aim at providing a software framework to CSP. The latter must use this framework, meaning that virtualization management is done by the provided framework. The latter replaces the software stack usually used by CSP. Provided framework allows users to specify different policies regarding their data. One of policies is about data location. Such a policy specifies which locations are allowed and which ones are forbidden, or more specifically which data centers are allowed and which ones are forbidden. Policies are interpreted by the framework before any operation on data; if the operation meets the policies it is executed, otherwise it is rejected.
  In case users do not trust CSP, Cloud Framework-based approaches can be combined with some hardware root of trust, administered by a third party, to ensure the framework is really run on the CSP. These approaches provide users with location guarantees through the installed framework regarding data location, but they are not, strictly speaking, verification methods.
- Hardware-based approaches [18–20] aim at providing a tamper-proof hardware root of trust, attached to one or several CSP's physical machines that guarantees their own locations, so by linking such a hardware to data it can guarantee data location. Such a dedicated hardware can be remotely accessed by users when data location verification in required. On user's verification request, the dedicated hardware makes it sure data are within its scope and replies with its location. Based on received location from dedicated hardware, the user decides whether the location is suitable for his/her needs.
  In these approaches, hardware root of trust should be administered by a third party, trusted by the user and the CSP, in order to ensure correct usage and validity of location verification results. Meaning that the third party sets up the hardware, maintains it, plans regular audits to check hardware presence in the right locations and detect any misbehavior of CSP, i.e. when the CSP would move data to another location or succeed in breaking tamper-proof protections.
- Landmark-based approaches [10, 21–27] aim at providing communication-based solutions to estimate data location. Unlike previous approach classes, landmark-based approaches are not restrictive for CSPs, as they do not require installation of a specific hardware or software on CSPs. The user deploys landmarks in different known locations, trying to surround locations in which data are believed to be stored (see Fig. 2). Landmarks are ordinary servers with appropriate computing power and connected to Internet. In a first step, landmarks interact with each other to build a distance model, mainly based on Round-Trip Times (RTTs) between them. This step is a training step and results in either a regression model or a classification model. Then, on user's location verification request, landmarks interact with CSP where data are assumed to be stored to collect RTTs involving CSP. Using the previously built learning model and new RTT measurements, a geographic zone reflecting RTTs is derived. CSP location should be included in the derived zone, otherwise a malicious or accidental move of data to another location has occurred.

In the sequel, we only address landmark-based approaches for their flexibility and low cost. A review and a classification of landmark-based data location verification approaches in the cloud are proposed. Classification is based on identification of criteria the most significant to capture differences in the design of the most referenced approaches. Their experimental results also are introduced. Shifting data from valid location, which is the one of CSP location included in the SLA, may result from an accidental or malicious behavior of the CSP. In case of malicious CSP, location verification approaches should consider attacks coming from CSP to prevent users discovering that their data have been moved elsewhere. We identified potential attacks and methods to avoid or detect them. It should be noticed that, for performance reasons, including data access delay and robustness, data may be stored at different locations by CSP and users are aware of the distribution or duplication of their data. In such a case, location verification process is designed to verify a set a locations and not a single one. Without loss

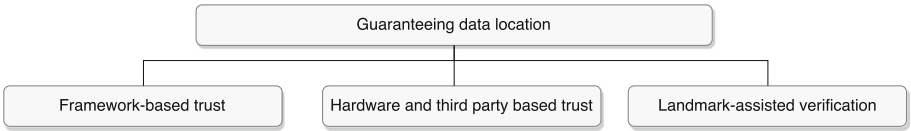

Fig. 1 Data location verification approach classes

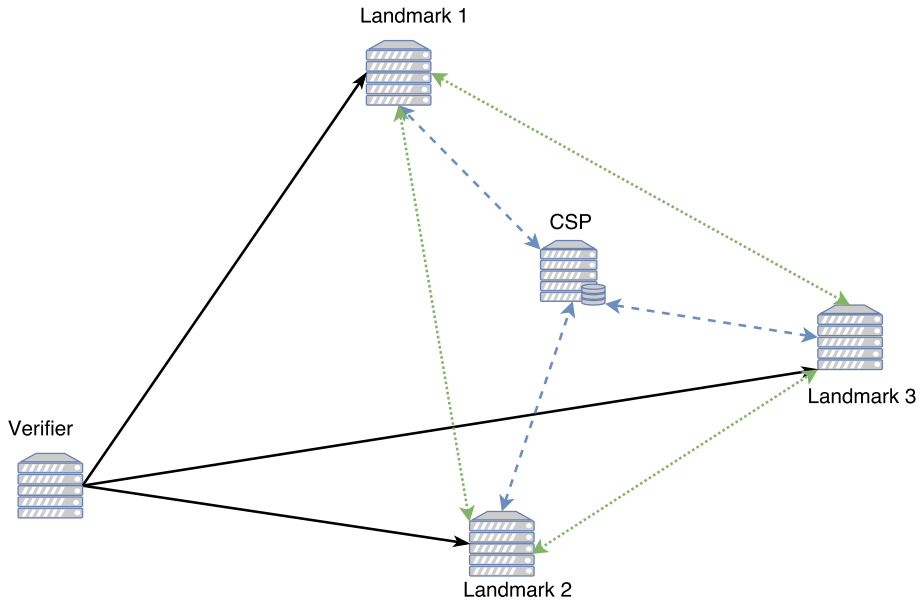

Fig. 2 Structure of landmark-assisted verification approaches. Solid arrows represent verification monitoring data flow, dotted ones the inter-landmarks measurement data flow, and dashed ones the measurements data flow between CSP and landmarks

of generality, a single location is assumed in the sequel. Iterating the verification process described in the following sections would contribute to consider multi-location CSPs.

Paper organization

The rest of this paper is organized as follows. The following section presents our classification of location verification approaches and their design models. In "Synthesis of experimental results" section, we highlight the main results derived from the experimentations carried out by authors of reviewed approaches. In "Potential attacks on location verification process" section, the potential attacks on the verification process are described. "Discussion and challenges" section discusses some challenges and "Conclusion" section concludes the paper.

Classification of landmark-based data location verification approaches

Location verification approaches are machine learning (ML) based. Roughly, the idea is that if we accurately learn about network performance regarding the zone where CSP is assumed to be, without CSP's participation in learning step, then when the CSP is probed, experienced network performance should be close or identical to the ones observed during learning, otherwise the CSP should be declared out of zone. More specifically, there are two steps in location verification approaches:

- *Training step*: landmarks interact with each other to collect network measurements including Round Trip Times (RTTs), number of hops, and so on. In the sequel, unless stated otherwise, "network measurements" mean RTT values. Then, measurements are used to compute parameters of a ML model, which is used in a second step to estimate a geographic zone associated with CSP.
- *Verification step*: when the user needs to verify CSP's location, landmarks are notified and then they interact (sending Ping requests or accessing the data stored on CSP) with CSP to collect RTT measurements involving the CSP. New measurements and the ML model established during training step are used to derive a zone in which the CSP is estimated to be located.

Existing approaches differ according to multiple design criteria. In the sequel, we present our classification, which aims at highlighting design specificities of the approaches. As summarized on Table 1, identified design criteria are grouped into five categories: correlation between proxy IP address and data locations, landmark involvement in the verification process, measurements collecting, machine learning, and use of PDP (Proof of Data Possession) protocol.

Definitions:

- ***Data location***, or ***location*** in short, is the zone in which the CSP is expected to be located.
- ***Location verification process***, or ***verification process*** in short, is the set of actions, including data

Table 1 Classification of landmark-assisted data verification approaches

	Correlation[a]	Landmark			Measurements collecting		PDP algorithm
	location / PIP@	Type	Distribution scale	Selection	Metrics	Probing protocol	
Biswal2015 [27]	Yes	Active	USA	None[c]	RTT, HC[d], BW[e]	ICMP	None
Ries2011 [24]	Yes	Active	Worldwide	None[c]	RTT	ICMP	None
Fotouhi2015 [10]	Yes	Active	Worldwide	Pre-verification	RTT	ICMP	None
Jaiswal2016 [22]	No	Active	USA	Pre-verification	RTT	ICMP, HTTP	None
Benson2011 [25]	No	Active	USA	None[c]	RTT	HTTP	None
Gondree2013 [26]	No	Active	USA	None[c]	RTT	HTTP	MAC-based
Watson2012 [21]	No	Active	USA and Europe	Pre-verification	RTT	HTTP	MAC-based
Eskandari2014 [23]	No	Passive[b]	Worldwide	Pre-training	RTT	HTTP	None

	Machine learning			
	Training coordination	Distance estimate	Location inference	Location granularity
Biswal2015 [27]	Centralized	None[f]	Naive Bayes classification	County
Ries2011 [24]	Centralized	Virtual network coordinates	Instance-based Classification	Country
Fotouhi2015 [10]	Decentralized	Bestline	Multilateration	Area[g]
Jaiswal2016 [22]	Decentralized	Distance to delay ratio	Multilateration	GPS coordinates
Benson2011 [25]	Centralized	Linear regression	Hierarchical clustering	City
Gondree2013 [26]	Decentralized	Bestline	Multilateration	Area[g]
Watson2012 [21]	Decentralized	Linear regression	Multilateration	Area[g]
Eskandari2014 [23]	Centralized	Polynomial regression	Multilateration	GPS coordinates

[a] Correlation between data location and proxy IP addresses
[b] This solution deploys only passive landmarks and measurements are performed by a virtual machine located on CSP
[c] None: all the landmarks in the initial set are used in the whole verification process
[d] HC: number of hops on route
[e] BW: bandwidth of route
[f] The classifier returns a location rather than a distance
[g] The size of the area depends on measurements

collecting, training, and location inference performed by participating nodes.

- ***Probing node*** is any participating node, which collects network metrics through probing requests.
- ***Probing request*** is any mechanism to collect network metrics, including RTT. *Ping*, *Traceroute*, and data access requests are the main forms of probing requests.
- ***Verifier*** is the node, which coordinates verification process and takes the final decision regarding data location.

Correlation between Proxy IP address and data server location

As mentioned in "Introduction" section, data owners are aware—through SLA—of data server location(s) and IP address(es) to access data. Reviewed location verification approaches differ in the way they correlate IP addresses to data server locations:

- *With correlation*: IP addresses included in SLA are either those of data servers or they are associated with proxies close to data servers. In other words, proxies enabling data access are in the same location than data servers. Consequently, when location verification is of concern, probing CSP's proxy is equivalent to probing CSP's data server.
- *Without correlation*: provided IP addresses are not correlated to data server locations. Proxies may be very far from data servers. Thus, their locations are without help to locate data servers.

It is worth noticing commercial CSPs, which commonly rely on virtualization, are more likely to follow the second alternative to make their infrastructure transparent regarding users and also to protect it from attacks.

Landmarks

A landmark may be any host connected to Internet and

whose physical location is known to other landmarks and to the Verifier. Landmarks collaborate to estimate CSP location compared to their own locations. Location Verification approaches differ according to tasks assigned to landmarks.

Landmark type

Landmarks can be categorized depending on their participation in the verification process. Deployment of location verification may involve two types of landmarks: *active* and *passive*. Active landmarks initiate measurements, collect RTT values, and may follow a learning process and derive distance from RTT values. Passive landmarks only reply to probing requests when they are solicited by active landmarks to collect RTT values. All approaches analyzed in this paper use active landmarks except the one proposed in [23], which deploys only passive landmarks and measurements are performed by a virtual machine located on CSP.

It is worth noticing that on one hand distance accuracy and zone estimate depends on collected data. The more landmarks are deployed, the more measurements are collected, and the more accurate are estimates. However, the verification process should be kept at a reasonable cost and the number of active landmarks, which agree to collaborate, is limited on the other hand.

Landmark distribution scale

Landmark distribution scale refers to the area in which landmarks are located. Such an area may be the Earth, many continents, a single continent, a very large country, a small country, or a part (i.e. a state, a region, a county...) of a country. In [22, 25–27] continental scale is used, deploying landmarks in the USA [21] also uses a continental scale, as landmarks are located in Europe and the USA. [10, 23, 24] use worldwide scale.

It is worth noticing that, with a reasonable number of landmarks and verification cost, the wider is the landmark area, the lower is the accuracy of distance or zone estimate. For example, in Canada or Russia, with a radius of 2000 km we are still in the same country, while with the same radius, many countries are covered elsewhere in the world. Thus, landmark distribution scale is dependent on location granularity requirements of users (i.e. a single country, a state...) and acceptable cost.

Landmark selection

Measurements collected by landmarks should contribute to derive the zone of CSP location. However, all landmarks could not provide the same contribution. For example, a landmark too far from the others or from CSP will at best contribute to increase variance of measurements and at worst to derive a zone not including CSP. Landmark selection aims at finding a compromise between number and locations of landmarks to deploy and data location estimate accuracy. In the approaches analyzed in this paper, landmark selection may be used at two levels:

- *Pre-training landmark selection*: first, a set of nodes that may be used as active or passive landmarks is considered. This set may be static or provided by specific function (e.g., all academic websites in a country). It is worth noticing that authors focusing on location verification process do not take care of this set arguing it is out of the scope of location verification. However, a good choice of the initial set is a key for success and accuracy of the verification process. Then either all nodes in the initial set participate in the training step or only a subset is selected. Indeed, in order to minimize communication cost of training and optimize accuracy of zone estimate, a selection of the closest nodes is performed and only those nodes participate in training, as proposed in [23].
- *Pre-verification landmark selection*: after training step and when the user requests location verification, data collected by all trained landmarks and their distance models are considered. Proposed solutions suggest two design schemes: all collected data are used to estimate CSP's zone or only a subset is used. The first scheme is easier to implement, but its accuracy is strongly dependent on pre-training landmark selection. The second scheme is more adaptive and aims at discarding landmarks that would jeopardize the verification process or results in a too much wide zone [10, 22].

Coordination of participants' training

Landmarks are selected to participate either as active or passive. The latter do not collect anything. Active landmarks may have two roles: Collecting measurements and Distance estimate. When location verification is based on landmarks, which only collect measurements, the solution is *centralized learning-based*, because the selection of learning models to infer the location zone is made by the Verifier, which may be the user machine or another delegated host. For load balancing and robustness reasons, landmarks may collect data, learn, and estimate distances, resulting in *decentralized learning-based* solution. Figures 3 and 4 depict operations of centralized learning-based solutions and Figs. 5 and 6 operations of decentralized learning-based solutions.

Measurement collecting

Measurement collecting is the distinctive feature of landmark-based approaches compared to other location

guaranteeing approaches. As previously mentioned, measurements are collected to fit machine learning models. Proposed location verification solutions are categorized according to network metrics and how they measure them.

Metrics

Different metrics may be used to determine CSP's location. The most used metrics are relating to the RTT and include raw RTT values, mean, mode, median, and standard deviation of RTT. Hop count and bandwidth of route between probing landmark and target node may also be collected to enforce the accuracy of the learning model [27].

Probing protocol

Communication protocols involved in collecting measurements are useful to differentiate location verification solutions. Mainly two schemes have been proposed: 1) collect RTTs using mainly *Traceroute* or *Ping* commands, both based on ICMP protocol, or 2) using data accesses based on HTTP. The first scheme results in more accurate RTTs, because in the second scheme overhead and its variation, when data are really accessed on disks, raises more variance in observed RTTs. It is worth noticing that the second scheme is useful when the verification process collects RTTs during data accesses by applications, which minimizes communication cost of the verification process, and also when the prober wants to check that data are really on the responding server (see Attacks in "Potential attacks on location verification process" section). ICMP messages are used in [10, 27]. HTTP messages are used in [21, 23, 25, 26]. Both message types are used in [22].

Distance estimate

When network-related metrics, mainly RTTs, are collected, decentralized learning-based approaches use them to infer a distance between each probing landmark and CSP's location. Roughly, distance estimate is a function f that takes measurements M as input and returns a distance d: $d = f(M)$.

There are several ways to select f function:

- using linear regression [21, 25],
- using polynomial regression [23],
- using a delay to distance ratio, which corresponds to the average of measured delay between landmarks over distance between probed hosts [22],
- deriving a bestline, which is a line with the highest intercept and slope such that all measured values are above the line [10, 26].

It is worth noticing that each of the above ways to select estimate function has its pros and cons from the statistical analysis point view. Indeed, it is well known, in the statistical analysis field, that selection of a regression model and its parameters depends on characteristics of observed data and the estimate error bound. Selecting the optimal estimate function is still a challenge, which is out of the scope of this paper.

Instead of inferring a function giving Cartesian distance between a couple of nodes, a function mapping nodes to virtual coordinates according to measurements can also be considered. Virtual coordinate systems (VCS), such as Vivaldi, Pharos, and Phoenix, are schemes proposed to locate hosts in the Internet. Based on dedicated landmarks, VCSs map nodes into a geometric space in such a way that their distance in the VCS represents latency

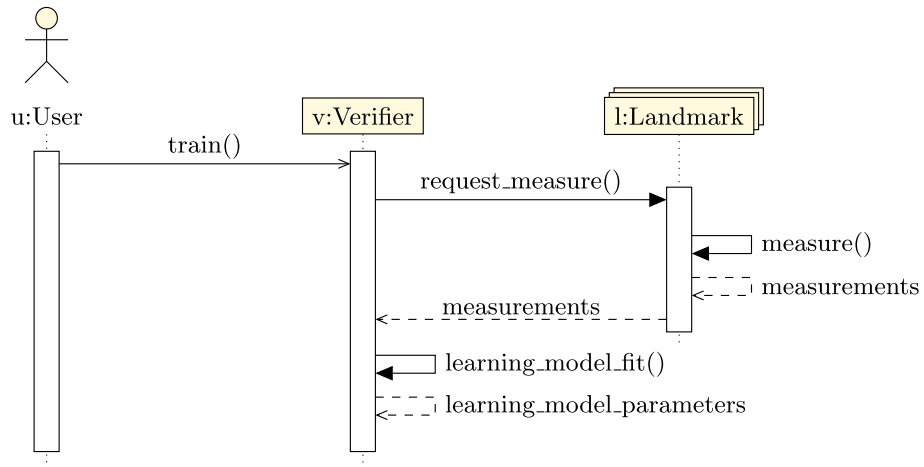

Fig. 3 Training step (centralized approach)

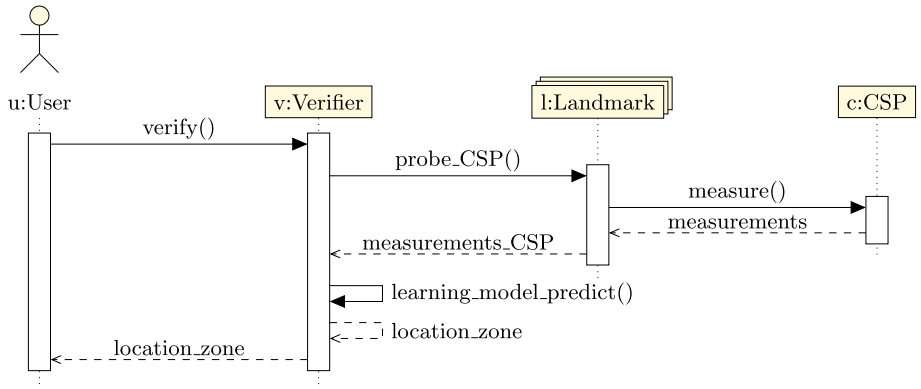

Fig. 4 Verification step (centralized approach)

between them in the physical network. Thus, distance between two hosts in the virtual coordinate space is a equivalent to RTT between these hosts. An example of approach based on VCS is proposed in [24], which uses Phoenix virtual coordinates system.

Depending on landmark training of the solution under consideration, a single distance estimate function is built by the Verifier using all collected data or each landmark builds its distance estimate function using its own collected data and then sends its estimated distance to the Verifier. The first scheme is called *centralized distance estimate* and the second *decentralized distance estimate*. Decentralized distance estimate functions are proposed in [23–25] and centralized ones in the other solutions except for [27] that does not include any distance estimate function. As far as we know, distance estimate functions built locally by landmarks differ only in their parameters and not in their forms, i.e. all distance estimate functions used by landmarks may be linear, polynomial and so on, but without mix.

Location inference

Location inference method

Location Verifier performs the last task, which is inference of a CSP zone based on learning step and measurements involving the CSP. Zone inference depends on learning coordination:

- Location inference in centralized learning-based approaches: classification is mainly used in these approaches. Inter-landmarks measurements are used by the Verifier to fit a classifier; it is the learning step. Then, measurements between landmarks and CSP are used to predict CSP's location zone. Different types of ML classification may be used, including Naive Bayes, Instance-based learning, and Hierarchical clustering.
- Location inference in decentralized learning-based approaches: multilateration is used in such approaches. Recall that prior to this final step, landmarks sent their distance estimates. Let n be the number of active landmarks, (x_i, y_i) the coordinates

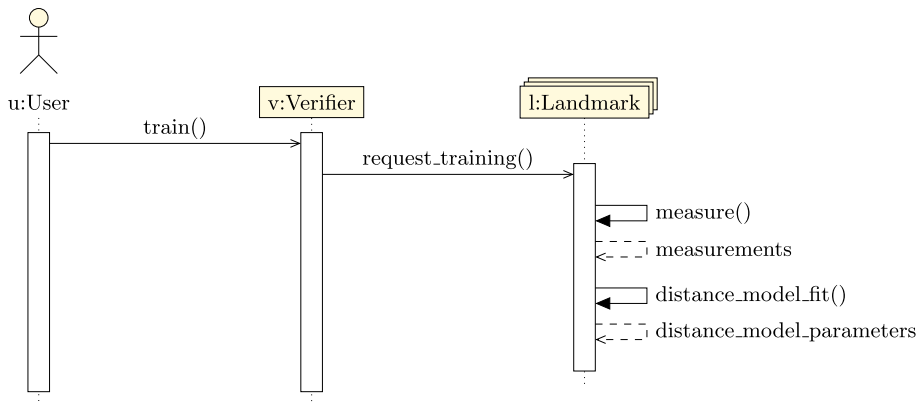

Fig. 5 Training step (decentralized approach)

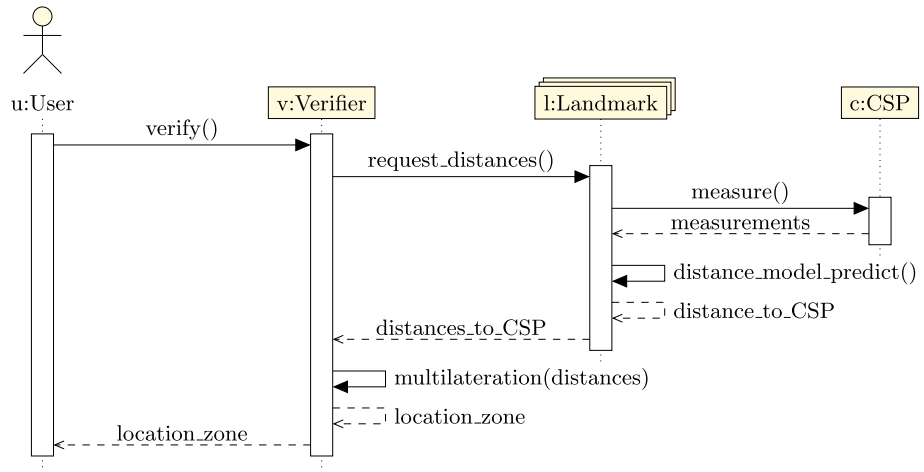

Fig. 6 Verification step (decentralized approach)

of landmark i, and d_i the estimated distance between landmark i and CSP. A circle, with (x_i, y_i) as center and d_i as radius, is associated with landmark i. Multilateration is the function which takes as input a set of n circles and returns a zone, which is a polygon with a maximum of n sides, formed by the intersection of those circles. Then, interpretation of the yielded zone may result in a city, a country, a continent, etc. Figures 7 and 8 show examples of multilateration result.

Among reviewed solutions, the ones proposed in [24, 25, 27] use classification and the others use multilateration.

Location granularity

At the end of location verification process, a result is returned to data owner. Very fine granularity is reached when the result is in the form of GPS coordinates [22, 23], a city [25] or a county [27]. Coarse granularity is reached when the result is a country [24] or a continent. Some solutions, including [10, 21, 26], provide a very variable granularity depending on variance in measurements; granularity may vary from very fine to coarse.

Location granularity is a quantitative property of location verification process. Not all users require the same granularity. Many users only need their data to be located in their country and do not want to pay too much to receive verification result. As location verification process is based on machine learning, all estimates come with errors, thus the probability that CSP is in the inferred zone grows with the zone size. Under this observation, a very fine location granularity constraint may jeopardize location verification process and provides false negative. In other words, location verification result interpretation would be "CSP is not in the authorized area" with a narrow zone, while it would be "CSP is in the authorized area" with a wider zone.

Proof of data possession protocol (PDP) utilization

PDP protocol is a protocol that allows data owners to verify that their data are actually stored on a data server. It consists of four main operations: data pre-processing, inquiry, response, and check. There are two main design schemes for PDP protocols:

- Message authentication code (MAC) based scheme: data owner pre-processes data, generating a tag T_i for each block B_i of data using a hash function $H()$. Then, tags are stored by data owner and data blocks are sent to data server to be stored. When data owner needs to verify that the data server has the data, it sends a PDP inquiry including a list of c randomly selected block numbers. Data server reads and sends the requested data blocks. Then, data owner computes tags for received data blocks and compares them to tags locally-stored to confirm or not the possession proof. It is worth noticing that MAC-based PDP is bandwidth consuming, depending on the number of blocks included in proof inquiries and the frequency of these inquiries. MAC-based scheme is used in [21, 26].
- Cryptography-based scheme: to avoid bandwidth consumption incurred by the previous scheme, one of the well-known solutions has been proposed in [28], which may be summarized as follows:
  - Data owner generates a private key and a public key. Then, it associates a tag T_i with each data block B_i—tag calculation is based on the private key—and it sends data blocks and

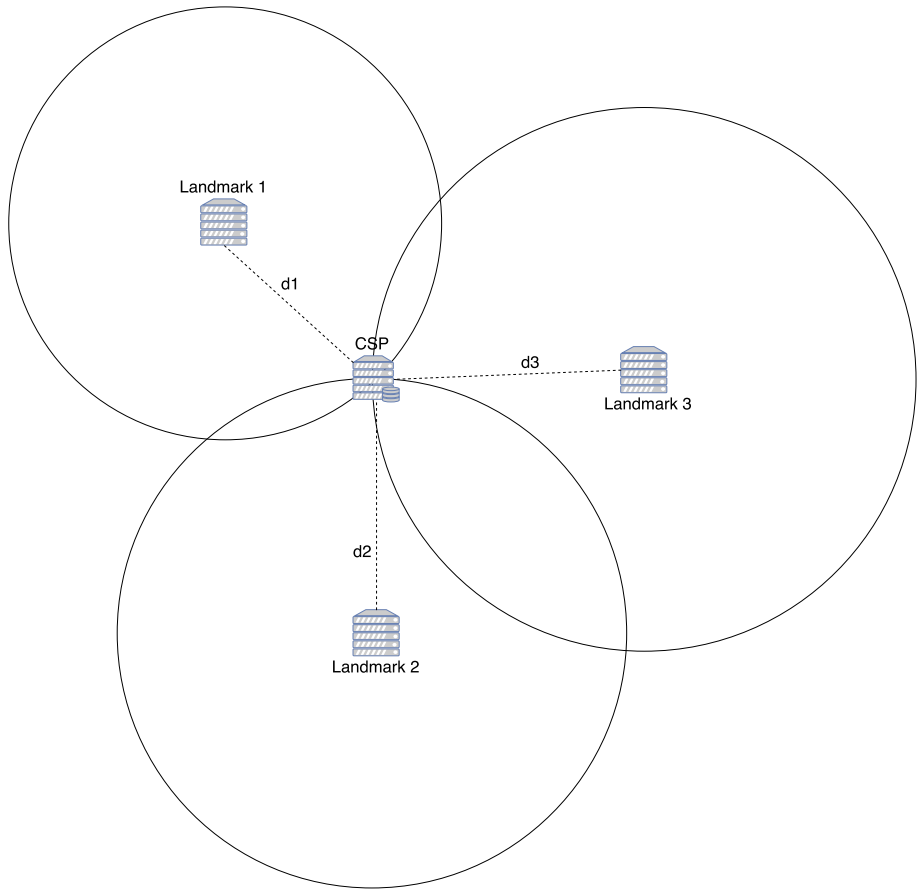

Fig. 7 Example of perfect multilateration. d_i the estimated distance between $Landmark_i$ and CSP is the same as the real one. Circles intersect at a point

associated tags to the server. Data owner may delete data and tags.
- When data owner needs to check the data server, it sends a PDP request including the public key and a challenge composed of a list of c data block numbers randomly selected and a random value r. Using random block numbers and a random value r prevents the server from anticipating which blocks will be queried in each challenge, and also prevents it from storing combinations (e.g., sums) of original blocks instead of the original file blocks themselves.
- Then, data server accesses c data blocks and uses the public key to generate a proof of possession composed of a tag and a hashed value of r. Notice that the data server cannot generate a valid possession proof without accessing the data file.
- Data owner receives the possession proof. Then it uses its private key to conclude whether the possession proof is valid or not.

As shown on Table 1, reviewed approaches don't use cryptographic-based PDP. We do recommend such a PDP scheme instead of MAC-based one as it is more robust and less resource consuming.

It is worth noticing that PDP protocols are mechanisms to provide guarantees that the responding server has the data, but do not provide guarantees on data location. Thus, usually PDP protocols are used jointly with other mechanisms, such as RTT collecting and checking, to provide a complete location verification.

Synthesis of experimental results

Papers addressed in this review include experimentations in order to assess performance of proposed location verification approaches under real scenarios. To provide comparison of analyzed approaches, distance error and verification success rate, as well as experimentation contexts, are taken from the original papers and summed up in Table 2. It should be noted that the presented results are the ones claimed by approaches' authors.

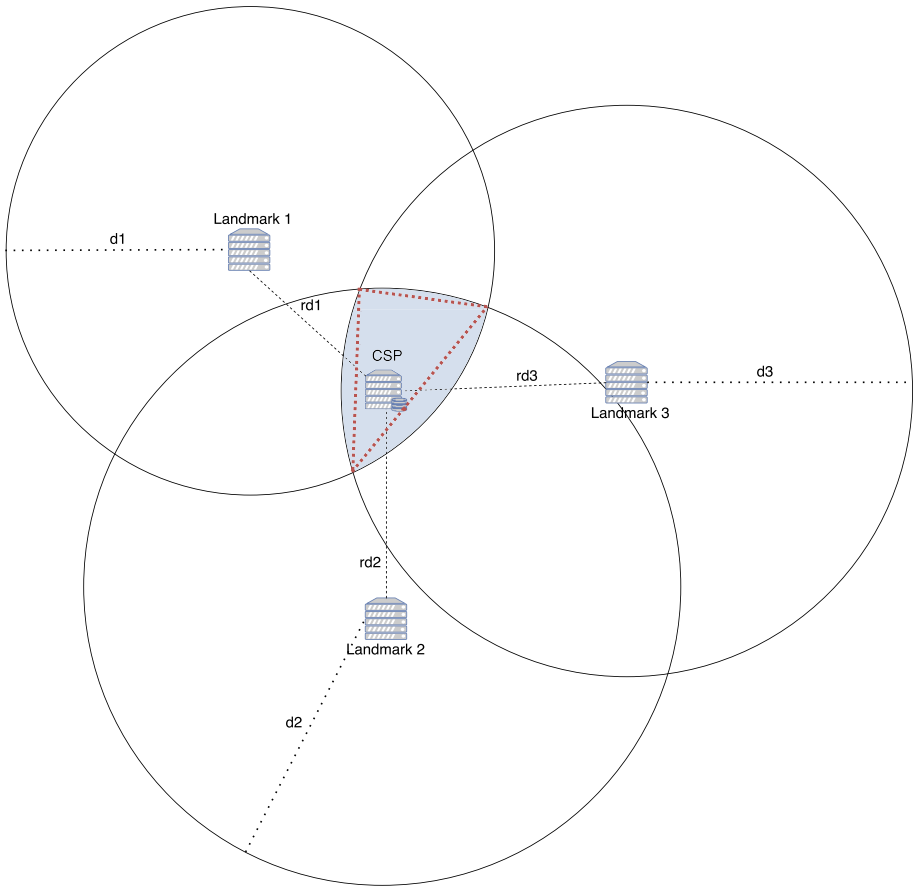

Fig. 8 Example of multilateration where returned zone is a triangle. d_i the estimated distance between $Landmark_i$ and CSP is different from rd_i the real one

Biswal et al.'s approach

Described experimentation was based on a set of 67 Planetlab nodes, acting as landmarks, which interacted with 23,843 US routers with known locations. Data files were uploaded on Amazon, Rackspace and Google cloud storages. For each storage service, the associated IP addresses were discovered by downloading files. Then, each landmark probed discovered IP addresses and collected measurements including RTT, hop count, and bandwidth. Results show that using standard deviation, mean, and hop count results in better accuracy than using the mean alone and that adding the mode and median values does not result in any accuracy improvement. In case of Amazon CSP, 95% of predictions have less than 1.6 km error and 100% have less than 64 km error. In case of Rackspace and Google, almost 100% of predictions have less than 1.6 km error and 100% is reached with respectively less than 2253 km error and 1609 km error. When bandwidth measurements are considered, 100% of Amazon predictions have negligible distance error.

Ries et al.'s approach

The authors used 80 to 89 Planetlab nodes acting as landmarks and distributed in 28 countries. Collected RTTs were used to build an VCS (Virtual Coordinate System) model. Three VCSs—namely, Pharos, Vilvadi, and Phoenix—and two classification algorithms—Instance-Based Learning and Support Vector Machine—have been compared. Each landmark acted as a CSP to collect RTTs and could use a proxy. The latter impacts distance error. RTT measures were used together with the VCS to compute a virtual position for the landmark acting as a CSP. The location was then estimated thanks to the classification algorithm and the virtual position of other landmarks. When associated with Phoenix VCS, the best classification algorithm is Instance-Based Learning. Support Vector Machine works better with Pharos and Vivaldi VCSs. Phoenix outperforms the other VCSs. Without a proxy, success rate is 90% for Phoenix, 80% for Pharos, and 65% for Vivaldi. With a proxy, success rate is 50% for Phoenix, 40% for Pharos, and 30% for Vivaldi. Thus, including a proxy has a significant impact on distance error improve-

Table 2 Experimental contexts and results of landmark-based data verification approaches

	Experimentation context					Results		
	Number of landmarks	Type of landmarks	Distribution scale	Used genuine CSP	Settings[b]	Success rate	Granularity	Distance error (km)
Biswal2015 [27]	67	Planetlab	USA	Amazon	Mean, std and HC	95.00%	County	1.6
						100%	County	64
				Rackspace		99.99%	County	1.6
						100%	County	2253
				Google		99.99%	County	1.6
						100%	County	1609
				Amazon	Mean, std HC and BW	100%	County	1.6
Ries2011 [24]	80 - 89	Planetlab	Worldwide	None[a]	Phoenix, no proxy	90%	Country	
					Pharos, no proxy	80%	Country	
					Vivaldi, no proxy	65%	Country	
					Phoenix, proxy	50%	Country	
					Vivaldi, proxy	40%	Country	
					Pharos, proxy	30%	Country	
Fotouhi2015 [10]	9	Amazon VM	Worldwide	Google		100%	Area[c]	240
Jaiswal2016 [22]	60	Planetlab	USA	None[a]		100%	GPS coord.	88.5
				Amazon		100%	GPS coord.	112.7
Benson2011 [25]	36	Planetlab	USA	None[a]		100%	City	441.6
Gondree2013 [26]	50	Planetlab	USA	None[a]		50%	Area of 171,819 km^2	166
						90%	Area of 1,960,510 km^2	626
				Amazon	Min RTT	100%	Area of 11,175 km^2	
					Median RTT	100%	Area of 243,791 km^2	
Watson2012 [21]	28	Planetlab	USA and Europe	None[a]	Europe	50%	Centroid coord.	800
						75%	Centroid coord.	1000
					North America	50%	Centroid coord.	1000
						75%	Centroid coord.	1200
Eskandri2014 [23]	38,892	Websites	Worldwide	None[a]	Landmarks at <100 km	100%	GPS coord.	100
					Landmarks at <1000 km	100%	GPS coord.	600

[a] None: a landmark simulating a CSP is used
[b] Settings: tested options in estimate functions, system architecture, and node deployment
[c] The area considered in the experimentation is the border between two US states

ment. However, the authors notice that increasing the number of landmarks does not necessarily improve the distance error beyond a threshold.

Fotouhi et al.'s approach

Nine Amazon virtual machines, in 9 different locations, were used as landmarks. RTT measurements were used to build a bestline for each landmark. A virtual machine has been deployed on Google cloud. Such a VM was assumed to be located in the USA, so landmarks outside the USA were ignored. Remaining landmarks sent requests to the VM to measure RTT, each one using its own function on measured RTTs to estimate its distance to the VM. Using estimated distances and a multilateration algorithm, a location has been derived. Provided results show a distance error of 240 km along the border between two US states. It should be noticed that a single landmark close to data server was chosen to reach such an error, which is the best one in experimentation.

Jaiswal & Kumar's approach

Sixty Planetlab nodes were selected to be used as landmarks. Two experimentations have been carried out, using a landmark simulating a CSP or an Amazon CSP. Reported distance error is 88.5 km in case of simulated CSP and 112.7 km for Amazon CSP. The increase in distance error is due the density of landmarks used in both experimentations. There were less landmarks around Amazon CSP.

Benson et al.'s approach

Thirty six Planetlab nodes were used. Three delay-to-distance estimate models (based on $\frac{4}{9}$ light speed, global bestline, and linear regression) have been tested. Using selected delay-to-distance estimate models, landmarks estimated their distance to 40 university websites—assuming universities host their websites. Experimentation results show that the global bestline based model outperforms other models and has an average distance error of 441.6 km.

Gondree & Peterson's approach

Fifty Planetlab nodes were chosen to be landmarks acting as CSPs. Minimum RTTs resulting from data block transfer between landmarks have been used to build a bestline estimate model. Then, local estimated distances have been used to apply a multilateration algorithm, which returned the estimated location.

Two experimentation results were reported, a first experimentation using a simulated CSP and a second using an Amazon CSP. In the first experimentation, results show that 90% of predictions locate the centroid within 626 km margin in an area of 1,960,510 km^2) and 50% within 166 km margin in an area of 171,819 km^2.

In the second experimentation, data blocks were stored on Amazon CSP. In case of using minimum RTT values, the best obtained area surrounding the estimated location is 11,175 km^2. However, obtained area is 243,791 km^2 with median RTT values. No distance error was provided.

Watson et al.'s approach

Eighteen North American Planetlab nodes and 10 European Planetlab nodes were used as landmarks. Each landmark acted as a CSP and all the others sent 50 requests from which only the minimum RTT is kept to build a delay-to-distance function using linear regression. One experimentation deployed a server in Europe and another one in North America. Using estimated distances and multilateration algorithm, coordinates of zone centroid were predicted.

Provided results indicate that varying PDP parameters did not impact the error rate. In Europe, for 50% of prediction tests, the error was less than 800 km and for 75% tests, the error was less than 1000 km. In the USA, for 50% of tests, the error was less than 1000 km and for 75% it was less than 1200 km.

Eskandari et al.'s approach

A virtual machine on CSP runs location verification process. VM interacted with landmarks sending HTTP requests to measure RTTs. Among a set of 38,892 websites with known locations and located within 1000 km range, at least 500 were selected to build a delay-to-distance function using polynomial regression. Data server was located in Trento, Italy. Results show an increase of error with the range of landmark selection. When the range is 100 km, the error is 40 km; a higher range of 1000 km results in a 600 km error.

We would like to notice that the included results are ambiguous. Distance error value is not clear, because two graphs are provided. One graph shows a comparison with other approaches and the second one is a self-comparison with different parameter values. Unfortunately, reported results are not coherent. On the first graph, estimate error is 100 km with 1000 km range, while it is 600 km with 1000 km range on the second graph.

Potential attacks on location verification process

In order to tamper landmark-based location verification, a malicious CSP may implement different attacks depending on location verification approach. The goal of malicious CSP's attacks is to hide real data location. Instead of storing data in the agreed location, the CSP moves data to another location, possibly another country, where the user does not want his/her data to be stored and the CSP

tries to hide this fact to user. In such a case, the CSP is considered malicious and it deliberately implements specific attacks. Data movements would be undertaken to cut costs or for intelligence reasons. In the first case, CSP is economically rational: it would take some risks by moving data to an unauthorized location, but it does so only if storage cost is significantly reduced. The second case refers to various spying forms.

The main types of attacks regarding landmark-based location verification process are summarized in Fig. 9 and briefly presented below. Some attacks may be avoided by design; for example attacks on virtual machine are avoided when the user does not deploy any VM in location verification process. Some other attacks may only be detected; for example, RTT manipulation. Finally, some complex and/or costly attacks have not yet been considered in the literature.

Blocking verification

Attack principle

This basic attack is to prevent location verification process to complete; it is a type of denial-of-service. In this attack, access to CSP's resources is blocked for location verification process. There are two parts of location verification process that can be blocked by the CSP:

- Landmark blacklisting: CSP detects landmarks participating in location verification process and blacklists them because they are potential witnesses of malicious CSP's behavior. By preventing landmarks to collect RTT measurements, the verification process is blocked.
- Trace service blocking: the most common way to measure RTTs is to use ICMP queries. Under the pretext of security or performance reasons, CSP may decide to block ICMP queries. Any other protocol that is not the one enabled by the CSP to access data may also be blocked.

Solution

There are different solutions for this type of attack depending on what is blocked:

- Landmark blacklisting: there are three main solutions to avoid landmark blacklisting. The first is that user and CSP agree on a list of landmarks that will be used to verify data location. In such a case, CSP is aware of the existence of location verification and should avoid data off-shoring; the user is saying to CSP "I am watching you". The second one, which is used when the user wants to keep the verification secrete, is to activate multiple legitimate machines, which are authorized to access the data on CSP, and then collect RTT values. The third solution is to randomly select, at each training initialization time, landmarks among a very large set of nodes making it either infeasible or very costly for the CSP to blacklist all those nodes.
- Trace service blocking: to overcome trace service blocking, the user and some landmarks may collect RTT values while accessing data stored on CSP. It is assumed here that CSP cannot learn about the objectives of data accesses (i.e., when the user is really exploiting data or when data access is just an alibi to derive RTT values).

Using malicious proxy

Attack principle

CSP installs a proxy located at a user-authorized location included in SLA and stores data at another location. Then probing and data access queries are answered by the proxy, thus persuading the user that his/her data are at the right location. Such an attack is implemented differently depending on interactions between the verification process and the data access process:

- No interaction: location verification and data access processes are independent, i.e. the Verifier does not rely on data accesses to verify the location. In such a case, the verification process only considers network-related metrics, such as RTT, and the proxy only needs to reply to ICMP queries instead of the CSP. After a wave of RTT collecting what is estimated by location process is a proxy location, which is interpreted as valid location by the verifier resulting in a false positive. Later, when an access to data is requested, the query is transmitted by the proxy to the real data location node, which then sends requested data to the proxy, and then the proxy sends these data to the requesting node.
- With interaction: in this case, location process collects RTTs while accessing data. Thus, the proxy cannot maliciously behave as previously unless it deploys data caching or special high-speed connections, which are other attack types presented

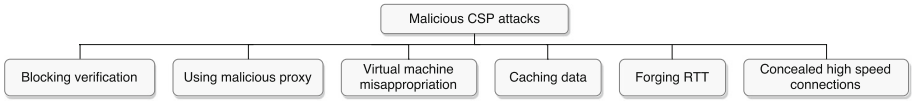

Fig. 9 Potential attacks on the verification process

later. Assuming no data caching and no special connection, the proxy may deploy attacks on location verification process if it is able to learn about data accesses and to classify them into two categories: i) data accesses used only for location verification process and the content of read data is not relevant, only RTTs matter, and ii) data accesses where content is relevant but no RTTs are collected. Under the assumption that such a learning is feasible, when a data request is received, the proxy needs to classify it as request type *i* or*ii*. When a request is classified as type *i*, the proxy forges a random data and sends it to the requesting node and when it is classified as type *ii*, a remote access to data server is made to receive authentic data, which are then sent to the requesting node. Consequently, location verification decision results in a false positive. It is worth noticing that this attack is potentially feasible only when the proxy is aware of the verification process details and data accesses have specific patterns (e.g., each first day of week location verification process is run).

Solution

Potential solutions to face this attack type are as follows:

- No interaction between the Verifier and Data access processes: to avoid proxy malicious behavior, a Proof of Data Possession (PDP) should be deployed to have guarantees that data are located on the responding proxy.
- With interaction: a first solution is to deploy a PDP as in the previous case. A second is to make the CSP unable to learn objectives of data accesses to detect those accesses used only to collect RTT measurements and those where the content matters for user's applications. This may be achieved by an appropriate sequencing of data queries.

Virtual machine misappropriation

Attack principle

In the cloud context, not only users' data are stored by the CSP but also user's programs may run on virtual machines (VM) hosted by CSP. In such a case, both data and location verifier are on CSP. It is worth noting that the user may choose to host, on the same CSP, totally or partially the tasks composing location verification process. Truthful CSPs run their clients' programs without any interference or attacks. However, when a CSP is malicious, it may force location verification tasks on its hosted virtual machines to send invalid data to derive current data location when the CSP has changed data location while not allowed by the user to do so. The VM may also be moved by a malicious CSP.

Solution

To avoid VM misappropriation by a malicious CSP, the simplest way is not to use VM to implement location verifier. Rather, location verifier should be hosted on a private machine owned by the user or on a trusted third party. However, if for any good reasons regarding user's preferences or requirements, location verifier is hosted by the CSP, the user must deploy on the CSP a trusted hardware, such as a Trusted Platform Module, to prevent CSP to manipulate the hosted VMs [29].

Forging RTT

Attack principle

RTT forgery is one of the basic attacks that may be used by a malicious CSP to obfuscate landmark-based location verification approaches. When location verifier tries to collect RTT values to derive CSP's location, it sends requests (Pings or data accesses) to CSP. Then, the latter delays or handles the request with a higher priority—which results in lower RTT values—so that collected RTT values either will not help the verifier to derive current CSP location or worst the verifier derives the agreed data location (i.e., the location included in SLA). It is worth noticing that decreasing RTT based attacks are much more complex than the ones that randomly increase the RTT. Two types of RTT forgery may be used by the CSP:

- Random RTT forgery: the amount of waiting time upon reception of a Ping or data request, in order to increase/decrease RTT, is randomly generated. This causes RTTs to appear totally uncorrelated to the distance between landmarks and CSP and location verification process fails, so data may be stored anywhere.
- Requester-location-aware RTT forgery: assuming the CSP has a certain knowledge on location verification approach and on locations and roles of landmarks, the waiting time to increase/decrease the RTT is forged depending on origin of the query (i.e., the landmark originating the request), so that data appear to be stored at the location included in the SLA and not at the current CSP location. By forging RTT values, the centroid of the zone where data are assumed to be is deliberately changed. Using this process, CSP may move the centroid where it wants.

Detection and solution

RTT forgery attacks can be detected or avoided depending on how RTTs are forged:

- Random RTT forgery: it might be reasonably assumed that CSP would increase/decrease RTTs only for verification queries, otherwise CSP outgoing

traffic will be impacted resulting in QoS degradation. Under this observation, location verifier may detect RTT forgery by comparing RTTs values measured by a set of selected landmarks around the CSP. Thus, RTT forgery detection may be implemented using cooperative RTT measurements and statistical learning on RTT samples to detect the forged random part of RTTs computed at different landmarks. Another way to detect RTT forgery is when the zone inferred from forged RTT measurements is too wide [30].

- Requester-location-aware RTT forgery: this attacks assumes the CPS is aware of landmarks' roles and positions and how RTTs values are used in location verification process to forge fake ones. To avoid this attack, a solution is to randomly select, at each training step initialization time, landmarks among a very large set of nodes, which makes it either infeasible or very costly for CSP to learn and deploy an RTT forgery for each participating landmark.

Caching data

Attack principle

This attack may be used to obfuscate the deployment of a PDP (Proof of Data Possession) protocol. Recall that PDPs are mechanisms to face malicious proxies. In caching data attack, CSP stores most of the data at a remote location while storing in its local cache data needed by location verification process. When a data access request is for location verification, CSP fetches data in its local cache, otherwise the request is forwarded to the remote node of data storage before sending a response to user. This attack is feasible only under the assumption that the CSP is able learn which parts of the data are used in the verification process in order to cache them and which parts are used by the conventional applications of the user in order to store them in another location. Notice that the data caching attack is different from the "With interaction" attack of a malicious proxy. In the latter, data used in the location verification are not relevant to the user so the proxy generates any data to answer the request, while in the caching attack data are used both by location verification process and by conventional user's applications.

Solutions

There are two alternatives to face this type of attack:

- Avoidance by design: to make the CSP unable (at a reasonable cost) to discover which parts of data are used in the verification, location verification process should associate an RTT with each data access request (whatever the use of data) and then use all RTTs to derive data location or randomly select data blocks to be used in location verification process.
- Detection: under the assumption that RTTs collected for cached data and other data are different—because when the CSP moves data it would result in RTT increase—, the Verifier may check the variability of RTTs between both sets of data and detect potential attacks.

Concealed high-speed connections

Attack principle

CSP is aware of the user's behavior, which shows that the user relies on RTT measured during data access to derive CSP location. CSP chooses a node at location X, using an appropriate high-speed network, such that delay access between CSP and the chosen node is negligible compared to the variation of RTT between the CSP and user. CSP stores data at the location X. Then, when the user sends data access requests to CSP, the latter reads data from location X and then forwards them to the user. It is worth noticing that this attack may also be used by a malicious proxy even though a PDP is deployed (see Malicious proxy attack).

Solution

As far as we know, there is no solution to thwart this attack, which is very costly to malicious CSPs. However, it can be noticed that distance between any locations is limited by the delay induced by network connection. Even with a private connection, the speed is limited to $\frac{2}{3}c$, with c the speed of light in vacuum. Considering this limit and necessary round-trip for a request between two nodes, moving the data of 1000 km would result in 10 ms increase of RTT. RTT overhead would probably be seen by location Verifier. Notice that 1000 km may not be a problem depending on the radius of the acceptable zone for storing data in a country or in a set of states...

Vulnerabilities of reviewed approaches

Vulnerabilities of reviewed approaches are summarized in Table 3, which shows the following:

- All approaches are vulnerable to landmark blocking, except [23], which uses a very large set of landmarks making it very costly to a malicious CSP to block the verification process.
- Approaches in [10, 24, 27] are vulnerable to trace service blocking and to malicious proxy without interaction, because they use ICMP requests to probe CSP. The other approaches are able to avoid both attacks as they use only data accesses to collect RTTs.
- Approaches in [22, 25] are vulnerable to malicious proxy attack with interaction, because they access data with HTTP to collect RTTs, but do not check the validity of received data. Approaches in [10, 24, 27] are not vulnerable to malicious proxy, because no

Table 3 Vulnerabilities to attacks of analyzed approaches

	Blocking verification		Malicious proxy		VM misappropriation	Forging RTT		Caching data	Concealed connection
	Landmarks	Trace service	No interaction	With interaction		Random	Requester-location aware		
Biswal2015 [27]	×	×	×			×	×	×	×
Ries2011 [24]	×	×	×			×	×	×	×
Fotouhi2015 [10]	×	×	×		×	×	×	×	×
Jaiswal2016 [22]	×			×		×	×	×	×
Benson2011 [25]	×			×		×	×	×	×
Gondree2013 [26]	×					×	×		×
Watson2012 [21]	×					×	×		×
Eskandari2014 [23]					×	×	×	×	×

×: the proposed solution is vulnerable to the attack; an empty case means no vulnerability

data accesses are made. Approaches in [21, 26] avoid such an attack owing to PDP protocol utilization. In [23], landmarks are passive and location verification is hosted on a VM of the CSP with trusted location, consequently, it has capabilities to avoid proxy attack.

- Approaches in [10, 23] are vulnerable to VM misappropriation, because they are partially (in [10]) or totally (in [23]) VM-dependent.
- All approaches are vulnerable to RTT forgery, to data caching attack—except [21, 26], owing to PDP protocol—, and to concealed connection attack.

In conclusion, reviewed location verification approaches are (very) far from being robust to prevent malicious CSP's attacks.

Discussion and challenges

Landmark selection

Most of existing approaches do not select landmarks except when building the initial landmark set, which is often considered out of the scope of location verification approaches. However, initial landmark set selection is of prime importance, as the quality of results directly depends on measurements provided by landmarks. One should not choose landmarks with too much randomness in the quality of network connection, otherwise measurements would experience high variance, resulting in a wider location zone. An exciting challenge for next generation location verification solutions would be the proposal of machine learning based approaches, which dynamically adapt landmark set according to the required granularity of data location and to observed network traffic conditions.

RTT-distance mapping function

Almost all of approaches are based on RTT-to-distance mapping functions assuming that:

1. RTT between two locations is proportional to the distance between those two locations.
2. The routing path between two locations approximately fits the direct geodesic distance between these locations.

Authors of solutions summarized in Table 1 present experimentation results to assess their models. However, they mainly consider specific scenarios in the USA where the previous assumptions might be realistic. Unfortunately, both assumptions are far from being realistic worldwide in most of cases where location verification may be of concern. For instance, we collected measurements during a long period (two months) between hosts in France and between hosts in France and in Germany, Canada, Africa, and USA. None of collected measurement sets has confirmed any of the assumptions. One significant example to be reported is relating to measurements between Toulouse (France) and hosts in Frankfurt and Berlin, *deutschland.de*, *goethe-university-frankfurt.de*, and *fu-berlin.de*. RTT regarding Goethe University's website is around 56 ms and it is around 65 ms for the second host in Frankfurt, while both hosts are in the same city. RTT is around 67 ms for the host in Berlin. While Berlin is over 400 km farther from Toulouse than Frankfurt, observed RTTs are nearly similar. The assumption that RTT is proportional to distance seldom holds and regression functions to estimate distances proposed in almost all analyzed solutions are too far from the reality of current Internet latencies. One challenge would be the proposal of learning approaches, which combine at least the network topology—including the number of hops between landmarks—and RTTs for each couple of hosts. Using the same estimate function and with the same parameters for all hosts would result in too much error from statistical point of view. Context-aware approaches are required

to enable different classes of users to accurately locate their data.

Location inference

Multilateration-based approaches

More than a half of analyzed approaches use multilateration jointly with a RTT-distance mapping function. Among those approaches, two consider the centroid point—i.e., GPS coordinates—of the area yielded by multilateration [22, 23]. The latter assumes "nearly perfect" estimate function and measurements. In case of inaccurate measurements, the resulting centroid would be invalid and location verification process inconclusive. Those approaches also are very vulnerable to RTT forgery attacks. The remaining multilateration-based approaches return a zone, so they could detect RTT forging attack. Assuming RTT is proportional to distance, the best representative value associated with the distance is the lowest RTT as proposed in bestline-based solutions [10, 26]. However, the minimum RTT value has a very low probability of occurrence when the current Internet is of concern. Using the lowest RTT value would result in an underestimate of distance either in the training step or in verification step, thus preventing the multilateration function to return a useful zone. Indeed, two behaviors may occur:

1. During training step the minimum value of RTT is not observed, while it is during verification step, resulting in a short predicted distance. In such a case, circles representing distance between landmarks and CSP do not intersect.
2. The minimum value of RTT is observed during training, while only high RTT values are observed during verification step. In this case, multilateration function would return a very wide zone, thus jeopardizing accuracy of verification result.

One challenge would be a statistical analysis of multilateration accuracy in large landmark sets with high variance in the RTT values.

Classification-based approaches

It is well known that classification techniques are more powerful in decision making, when they use a sufficient number of features to discriminate samples. Unfortunately, in location verification approaches, almost all classification methods use a single feature, which is RTT. When only RTTs are used in training step, at verification step, a set of collected RTT involving a target host H, may be associated with multiple classes—so multiple hosts may be predicted as target and not only host H—, because many probing hosts may experience similar RTT values. Indeed, multiple zones in the world or in a same country may have similar latencies. A first challenge would be the investigation of other metrics, in addition to RTT, number of hops and bandwidth, to better classify collected measurements and reduce overlap between classes. A second challenge would be to test in parallel multiple classification models and to select the best one, which dynamically better matches observed measurements. It is worth noticing that testing simultaneously different learning models, for the same training and testing data, is a well known practice in machine learning. Such a practice is not, as far as we know, sufficiently addressed in location verification approaches.

Proofs of data possession

It is worth noticing that without guarantees that any server replying to probing requests is the right one, collected RTT measurements are useless to location verification process. As shown in Table 1, only two solutions include Proof of Data Possession protocols. It is surprising that PDP protocols are not prevalent in existing approaches. As already outlined, PDP protocols provide powerful mechanisms to trust collected measurements regarding RTT. One challenge would be an analysis of PDP cost regarding frequency of PDP checks. In particular, adequately increasing frequency of PDP checks makes it economically unprofitable for a malicious CSP to move data between two PDP checks. Indeed, a malicious CSP, which is aware of the period between two successive PDP checks, when this period is very high—e.g., more than one month—would be tempted to move data to another location between two PDP checks. Doing so, a malicious CSP would bring back data just when a possession proof is to be built and then it relocates data.

Accuracy and granularity of location

First of all, let us recall that all the proposed location verification solutions are based on statistical approaches, which come with estimate errors, in particular due to variance in collected RTT values. Controlling errors of distance estimate functions and location decision functions is very challenging. Some users would like to know the finest granularity (i.e., the data center's GPS coordinates or address) with the highest accuracy (i.e., with a minimum error margin). Unfortunately, landmark-based location verification approaches are intrinsically less accurate than other methods of location guaranteeing, in particular hardware-based methods. So one should find a compromise in terms of granularity (e.g., a city, a country, and so on), accuracy within the admitted granularity (e.g., a few km, hundreds of km, and so on), and verification cost. A final challenge we would like to rise is to associate with each solution decision result, not only a location zone, but also probability of success to quantify how the yielded result should be trusted by users. The latter may

be happy with an answer such as "data are in London city with a margin of 100 km", but unhappy with an answer like "data are in London city with a margin of 100 km with a probability of 0.53". Location verification approaches we analyzed do not (or marginally) consider such a challenge.

Conclusion

Data location in the cloud is one of the primary concerns for cloud users and it became a challenge. Many solutions have been proposed to verify data location. This paper presents a survey of the most referenced landmark-based data location verification approaches, which are flexible, low cost, and not restrictive for CSPs, as they do not require installation of dedicate software of hardware in CSPs. First, we present a comprehensive classification criteria for those approaches. Identified criteria are grouped into five categories: correlation between proxy IP address and data location, landmark involvement in the verification process, measurements collecting, machine learning, and PDP protocol deployment. After classification of these approaches, experimental setup and results produced by their authors are detailed. Then, the criteria guide identification of potential attacks that a malicious CSP might implement to jeopardize the location verification process. Malicious CSP relocates data on data servers, which are not authorized by users, and then tries to make them believe their data are on the location agreed in SLA. We briefly describe some solutions to overcome attacks or detect them. Finally, we discuss limitations and drawbacks of existing solutions and identity some challenges that require further investigation in the future. Location verification still remains an open and exciting issue in the cloud computing research field.

Funding
Not applicable

Authors' contributions
All the authors contributed to the design and implementation of the research, to the analysis of the results and to the writing of the manuscript. Their names appear in alphabetic order. All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Received: 19 June 2017 Accepted: 14 November 2017
Published online: 29 December 2017

References

1. Amazon Elastic Compute Cloud (EC2). Accessed 16 JUNE 2017. https://aws.amazon.com/ec2
2. Google Compute Engine. https://cloud.google.com/compute. Accessed 16 Jun 2017
3. Heroku Cloud Application Platform. https://www.heroku.com/. Accessed 16 June 2017
4. Microsoft Azure Cloud computing plaftorm & services. https://azure.microsoft.com. Accessed 16 June 2017
5. Trello Project Management Application. https://trello.com/. Accessed 16 June 2017
6. GSuite by Google Cloud. https://gsuite.google.com. Accessed 16 June 2017
7. Canada Personal Information Protection and Electronic Documents Act. http://laws-lois.justice.gc.ca/eng/acts/P-8.6/. Accessed 16 June 2017
8. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data andrepealingDirective95/46/EC(GeneralDataProtectionRegulation). http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679. Accessed 16 June 2017
9. USA Health Insurance Portability and Accountability Act. https://www.gpo.gov/fdsys/pkg/PLAW-104publ191/html/PLAW-104publ191.htm. Accessed 16 June 2017
10. Fotouhi M, Anand A, Hasan R (2015) PLAG: Practical Landmark Allocation for Cloud Geolocation. In: 2015 IEEE 8th International Conference on Cloud Computing (CLOUD). IEEE. pp 1103–1106
11. AWS AWS Cloud Compliance. https://aws.amazon.com/compliance/eu-data-protection. Accessed 11 Oct 2017
12. Reuters BusinessNews November 17. https://www.reuters.com/article/us-russia-linkedin/russia-starts-blocking-linkedin-website-after-court-ruling-idUSKBN13C0RN. Accessed 16 June 2017
13. Service Level Agreement Standardisation Guidelines EuropeanCommission June 2014. https://ec.europa.eu/digital-single-market/en/news/cloud-service-level-agreement-standardisation-guidelines. Accessed 11 Oct 2017
14. Internet Protocol Address (IP) Geolocation Bibliography. http://www.caida.org/projects/cybersecurity/geolocation/bib/. Accessed 16 June 2017
15. Massonet P, Naqvi S, Ponsard C, Latanicki J, Rochwerger B, Villari M (2011) A Monitoring and Audit Logging Architecture for Data Location Compliance in Federated Cloud Infrastructures. In: 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW). pp 1510–1517
16. Betgé-Brezetz S, Kamga GB, Dupont MP, Guesmi A (2013) Privacy Control in Cloud VM File Systems. In: 2013 IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom), vol 2. IEEE. pp 276–280
17. Wüchner T, Müller S, Fischer R (2013) Compliance-Preserving Cloud Storage Federation Based on Data-Driven Usage Control. In: 2013 IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom), vol 2. IEEE. pp 285–288
18. Albeshri A, Boyd C, Nieto JG (2012) GeoProof: Proofs of Geographic Location for Cloud Computing Environment. In: IEEE (ed) 2012 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW). IEEE. pp 506–514
19. Krauß C, Fusenig V (2013) Using Trusted Platform Modules for Location Assurance in Cloud Networking. In: Lopez J, Huang X, Sandhu R (eds). Network and System Security: 7th International Conference, NSS 2013, Madrid, Spain, June 3-4. Springer Berlin Heidelberg, Proceedings. Berlin, Heidelberg. pp 109–121. Available from: http://dx.doi.org/10.1007/978-3-642-38631-2_9
20. Bartock M, Souppaya M, Yeluri R, Shetty U, Greene J, Orrin S, et al. (2015) Trusted Geolocation in the Cloud: Proof of Concept Implementation. Publication NISTIR 7904. National Institute of Standards and Technology, U.S. Department of Commerce
21. Watson GJ, Safavi-Naini R, Alimomeni M, Locasto ME, Narayan S (2012) LoSt: Location Based Storage. In: Proceedings of the 2012 ACM Workshop on Cloud Computing Security Workshop, CCSW '12. pp 59–70. Available from: http://doi.acm.org/10.1145/2381913.2381926
22. Jaiswal C, Kumar V (2016) IGOD: identification of geolocation of cloud datacenters. J Inf Secur Appl 27(Supplement C):85–102. Special Issues on Security and Privacy in Cloud Computing. Available from: http://www.sciencedirect.com/science/article/pii/S2214212616000168
23. Eskandari M, Oliveira ASD, Crispo B (2014) VLOC: An Approach to Verify the Physical Location of a Virtual Machine In Cloud. In: 2014 IEEE 6th International Conference on Cloud Computing Technology and Science (CloudCom). IEEE. pp 86–94

24. Ries T, Fusenig V, Vilbois C, Engel T (2011) Verification of Data Location in Cloud Networking. In: 2011 Fourth IEEE International Conference on Utility and Cloud Computing. IEEE. pp 439–444
25. Benson K, Dowsley R, Shacham H (2011) Do You Know Where Your Cloud Files Are? In: Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop, CCSW '11. pp 73–82. Available from: http://doi.acm.org/10.1145/2046660.2046677
26. Gondree M, Peterson ZNJ (2013) Geolocation of Data in the Cloud. In: Proceedings of the Third ACM Conference on Data and Application Security and Privacy, CODASPY '13. pp 25–36. Available from: http://doi.acm.org/10.1145/2435349.2435353
27. Biswal B, Shetty S, Rogers T (2015) Enhanced Learning Classifier to Locate Data in Cloud Data Centres. Int J Metaheuristics 4(2):141–158. Available from: http://dx.doi.org/10.1504/IJMHEUR.2015.074248
28. Ateniese G, Burns R, Curtmola R, Herring J, Kissner L, Peterson Z, et al. (2007) Provable Data Possession at Untrusted Stores. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, CCS '07. pp 598–609. Available from: http://doi.acm.org/10.1145/1315245.1315318
29. Achemlal M, Gharout S, Gaber C (2011) IEEE (ed) Trusted Platform Module as an Enabler for Security in Cloud Computing. In: 2011 Conference on Network and Information Systems Security. pp 1–6
30. Gill P, Ganjali Y, Wong B, Lie D (2010) Dude, Where's That IP?: Circumventing Measurement-based IP Geolocation. In: Proceedings of the 19th USENIX Conference on Security, USENIX Security'10. USENIX Association, Berkeley. pp 16–16. Available from: http://dl.acm.org/citation.cfm?id=1929820.1929842