On the vulnerabilities of landmark-based data location approaches: threats, solutions, and challenges

Malik Irain, Jacques Jorda, Zoubir Mammeri

On the Vulnerabilities of Landmark-based Data Location Approaches: Threats, Solutions, and Challenges

Malik Irain, Jacques Jorda, Zoubir Mammeri
IRIT, University of Toulouse, Toulouse, France
{malik.irain,jacques.jorda,zoubir.mammeri}@irit.fr

***Abstract*—Nowadays, many companies, administrations, and individuals are outsourcing the storage of their data to large-scale Cloud Service Providers (CSPs). Unfortunately, the modern cloud infrastructure virtualization results in the difficulty or the impossibility for data owners to know the location where their data are stored. 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. 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 the data in another location, like another country, in order to cut the costs or for any other reasons.**

This paper considers landmark-based verification approaches, which are flexible and low cost compared to other location verification approaches. The objective is the identification of vulnerabilities relating to the main proposed landmark-based location verification approaches when malicious CSPs are involved and the proposal of some countermeasures against CSP's attacks. We present, to the best of our knowledge, the most comprehensive literature survey on vulnerabilities of landmark-based data location approaches.

***Keywords*-Cloud computing; Data location verification; Landmark-based approaches; Threats**

I. Introduction

Cloud usage is growing every day and many companies, administrations, and individuals are outsourcing the storage of their data to large-scale distributed storage systems. Such users are thus relieved of the tasks related to the management and maintenance of the equipment for storing their data. The counterpart is that they lose some control on their data and they have to trust their CSP. For the Cloud to be more widely accepted, users can enforce their requirements through QoS clauses including the data location. Data location requirements are due to multiple aspects including legal issues [2], privacy [15], and performance [12].

Outsourcing the storage of user's data to the Cloud appears as a good solution. However, data storage is often subject to restrictions regarding the location of data. For instance, in Europe the governments force some data to be stored by certified companies in certified data centers [2]. The data storage contract must specify the data location. These restrictions also are applied in Canada [1], the USA [3], and many other countries. For all these reasons, users would like to be able to verify their data location. Unfortunately, because of today's cloud infrastructure virtualization, data owners cannot easily know the location where their data are stored. In addition, even in case of the establishment of a strong Service-Level Agreement (SLA), which includes an initial guarantee regarding data location, the CSP may then move the data to another location, like another country, in order to cut the costs or just by mistake.

During the SLA establishment phase, the CSP agrees on the data location specified through location clause. Then, the users have two alternatives: either trust the CSP or do not trust it and deploy appropriate mechanisms to verify the data location at any time. To consider data location verification, 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. Three location verification approaches classes are commonly distinguished:

- Cloud framework-based approaches [9], [17], [20] aim at providing a software framework to install on the CSP. Such a software is in charge of the data location verification.
- Hardware-based approaches [5], [7], [16] 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 the data it can guarantee data location.
- Landmark-based approaches [8], [10], [11], [12], [14], [15], [18], [19] aim at providing communication-based solutions to estimate data location. A landmark may be any host connected to the Internet and whose physical location is known. Unlike the previous approach classes, the landmark-based approaches are not restrictive for the CSP, as they do not require the installation of any specific hardware or software on the CSP. The user deploys landmarks in different known locations, trying to surround the locations in which data are believed to be stored. In a first step, landmarks interact with each other to build a distance model, mainly based on the Round-Trip Times (RTTs) between them. This step is the training step and results in machine learning model. Then, upon user's location verification request, landmarks interact with the CSP where data are assumed to be stored to collect RTTs involving the CSP. Using the previously built learning model and the new RTT measurements, a geographic zone reflecting the RTTs is derived. The CSP location should be included in the derived zone, otherwise a malicious or accidental move of the data to another location has occurred.

In the sequel, we only address landmark-based approaches for their flexibility and low cost. Shifting the data from the valid location, which is the one of the CSP included in the SLA, may result from an accidental or malicious behavior of the CSP. In case of malicious CSP, the verification approaches should consider attacks coming from the CSP to prevent them discovering that the data have been moved elsewhere to cut costs.

We identify the potential attacks and suggest methods to avoid or detect them. It should be noticed that for performance reasons, including data access delay and robustness, the data may be stored at different locations by the CSP and the users are aware of the distribution or duplication of their data. In such a case, the location verification process is designed to verify a set a locations and not a single one. Without loss 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. The objectives of the paper are the identification of vulnerabilities relating to the most cited landmark-based location verification approaches and the proposal of some countermeasures to elude malicious CSP's attacks. It is worth noticing that the paper is limited to presentation of attack countermeasures. Given the number of identified attacks, the analytical modeling of the proposed countermeasures and their simulation or experimentation are not addressed. Also, the proposed attack identification considers the chain of functions composing the verification process and how each function may be manipulated by malicious CSPs. Our goal is to answer the question "how to make the data location verification process as much robust as possible?" and not the question "how one can design a malicious CSPs?"

In the following sections, ***data location***, or ***location*** in short, means the zone in which the CSP is expected to be located. ***Location verification process***, or ***verification process*** in short, means the set of actions, including data collecting, training, and location inference performed by the participating nodes.

The rest of this paper is organized as follows. Section II presents the design characteristics of existing data location verification approaches, which are the most cited in the literature. In Section III, the potential attacks on the verification process are described and some countermeasures are proposed. It should be noticed that—because of space reasons—only the main ideas of the countermeasures are described. Section IV highlights the vulnerabilities of the selected data location verification approaches. Section V concludes the paper.

II. Landmark-based Data Location Verification Approaches

Recall that a landmark may be any host connected to the Internet and whose physical location is known to the other landmarks and to the **Verifier**. The latter is a node, which coordinates the verification process and takes the final decision regarding data location. Landmarks collaborate to estimate the CSP location compared to their own locations. 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 (state, region, county...) of a country. In [8], [10], [14], [15], [19] continental scale is used, deploying landmarks in the USA or in Europe. [11], [12], [18] use the worldwide scale. It is worth noticing that on one hand the accuracy of the distance and zone estimate depends on the collected data. The more landmarks are deployed and the more measurements are collected, the more accurate are the 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.

Most of location verification approaches are machine learning (ML) based. Roughly, the idea is that if one accurately learns about the network performance regarding the zone where the CSP is assumed to be, without CSP's participation in the learning, then when the CSP is probed the network performance experienced should be similar to the ones observed during learning step, otherwise the CSP should be declared out of zone. More specifically, there are two steps in the 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 the parameters of a ML model, which is used in the second step to the estimate a geographic zone associated with the CSP.
- *Verification step*: when the user needs to verify the CSP location, landmarks are notified and then they interact (sending Ping requests or accessing the data stored on the CSP) with the CSP to collect RTT measurements involving the CSP. The new measurements and the ML model established in the training step are used to derive a zone in which the CSP is estimated to be located.

The main building blocks of location verification approaches are shown on Figure 1 and described in the sequel. Existing approaches, which are summarized in Table I, differ on how the building blocks are designed and deployed.

A. Measurement Collecting

Measurement collecting is the distinctive feature of landmark-based approaches compared to the other location guaranteeing approaches. As previously mentioned, measurements are collected to fit machine learning models. Proposed solutions differ according to the network metrics and how they measure them.

Different metrics may be used to determine the CSP's location. The most used metrics are relating to the RTT and include raw RTT values, the mean, the mode, the median, and the standard deviation of RTT. Hop count may also be used to enforce the accuracy of the learning model [10]. Mainly, measurements are achieved through *Traceroute* or *Ping* requests or using data accesses based on HTTP. The first scheme results in more accurate RTTs because in the second

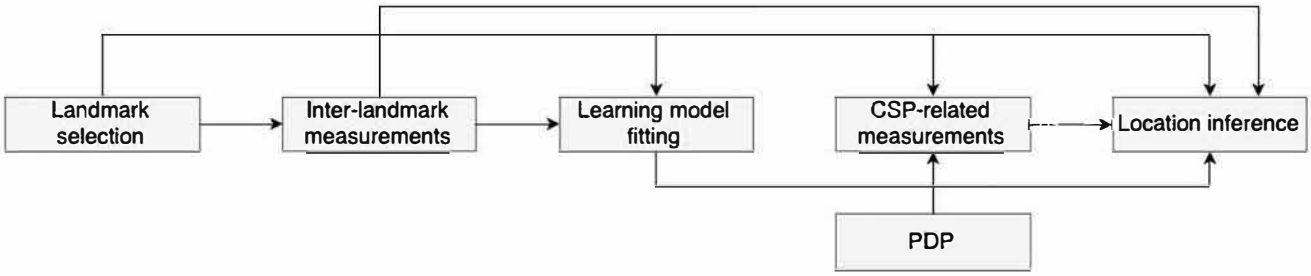

Fig. 1. Building blocks of location verification approaches (the arrows show different ways in combining the building blocks)

scheme the overhead and its variation, when the data are really accessed on disks, raises more variance in the observed RTTs. It is worth noticing that the second scheme is useful when the verification process collects RTT during data accesses by applications, which minimizes the communication cost of the verification process, and also when the prober wants to check that the data are really on the responding server (see Attacks in section III). *Traceroute* or *Ping* messages are used in [10], [12]. HTTP messages are used in [8], [11], [14], [19]. Both message types are used in [15].

B. Distance Estimate

Once network-related metrics are collected, decentralized learning-based approaches use them to infer a distance between each probing landmark and the CSP location. Roughly, the 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: linear regression [8], [19], polynomial regression [11], delay to distance ratio [15], and bestline [12], [14]. It is worth noticing that each of the above options to select the estimate function has its pros and cons from the statistical analysis point view.

Depending on landmark training of the solution under consideration, a single distance estimate function is built by the Verifier using all the 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 [8], [11], [18] and centralized local ones in the other solutions except for [10] that does not include any distance estimate function. As far as we know, the distance estimate functions built locally by landmarks differ only in their parameters and not in their forms, i.e. all the distance estimate functions used by landmarks may be linear, polynomial and so on, but without mix.

C. Location Inference

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

- Location inference in centralized learning-based approaches: mainly, classification is used in these approaches. Inter-landmark measurements are used by the Verifier to fit a classifier; it is the learning step. Then, measurements between landmarks and the CSP are used to predict the CSP location zone. Different types of ML classification may be used: Naive Bayes, Instance-based learning, and Hierarchical clustering.
- Location inference in decentralized learning-based approaches: multilateration is used in these 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 of landmark i, and d_i the estimated distance between landmark i and the 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 the zone, which is a polygon with a maximum of n sides, formed by the intersection of those circles. Then, the interpretation of the yielded zone may result in a city, a country, a continent, etc.

Among the analyzed solutions, the ones proposed in [8], [10], [18] use classification and the others use multilateration.

D. Proof of Data Possession protocol (PDP) Utilization

PDP protocol is a protocol that allows data owner to verify that data are actually stored on the 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: the data owner pre-processes the data, generating a tag for each data block using a hash function. Then, the tags are stored on the data owner and the data blocks sent to the data server to be stored. When the data owner needs to verify that the data server has the data, it sends a PDP inquiry including a list of randomly selected block numbers. The data server reads and sends the requested data blocks. Then, the data owner computes the tags for the received data blocks and compares them to the tags initially locally-stored to confirm or not the possession proof. It is worth noticing that MAC-based PDP is bandwidth consuming, depending on the number

TABLE I
MAIN DESIGN CHARACTERISTICS OF EXISTING LANDMARK-BASED LOCATION VERIFICATION APPROACHES

	Landmark selection	PDP algorithm	Machine learning		
			Training coordination	Distance estimate	Location inference
Biswal2014 [10]	None [1]	None	Centralized	None[2]	Naive Bayes Classification
Ries2011 [18]	None [1]	None	Centralized	Virtual network coordinates	Instance-Based Classification
Fotouhi2015 [12]	Pre-verification	None	Decentralized	Bestline	Multilateration
Jaiswal2015 [15]	Pre-verification	None	Decentralized	Distance to delay ratio	Multilateration
Benson2011 [8]	None [1]	None	Centralized	Linear regression	Hierarchical clustering
Gondree2013 [14]	None [1]	MAC-based	Decentralized	Bestline	Multilateration
Watson2012 [19]	None [1]	MAC-based	Decentralized	Linear regression	Multilateration
Eskandari2014 [11]	Pre-training	None	Centralized	Polynomial regression	Multilateration

1: All landmarks in the initial set are used in the whole verification process.
2: The classifier returns a location rather than a distance.

of blocks included in proof inquiries and the frequency of these inquiries. MAC-based scheme is used in [14], [19].

- Cryptography-based scheme: to avoid bandwidth consumption incurred by the previous scheme, one of the well-known solutions has been proposed in [6], which may be summarized as follows:
  - the data owner generates a private key and a public key. Then, it associates a tag with each data block—the tag calculation is based on the private key—and it sends the data file and the tags to the server.
  - When the 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 data block numbers randomly selected and a random value. Using random block numbers and the random value prevents the server from anticipating which blocks will be queried in each challenge, and also prevents it from storing combinations of the original blocks instead of the original file blocks themselves.
  - Then, the data server access the requested data blocks and uses the public key to generate a proof of possession composed of a tag and a hash from the random value.
  - The 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 I, 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.

III. POTENTIAL ATTACKS ON THE VERIFICATION PROCESS

In order to tamper with the landmark-based verification, a malicious CSP may implement different attacks depending on the verification approach. The goal of these attacks is to hide the real data location. In such a case, the CSP is considered as being malicious and it deliberately implements specific attacks. It is assumed that data off-shoring is made to cut costs, the CSP is economically rational: it would take some risks by moving data to an unauthorized location, but it does so only if the storage cost is significantly reduced. The main types of attacks regarding landmark-based location verification process are summarized in Figure 2 and presented below. Some attacks may be avoided by design; for example attacks on virtual machine may be avoided when the user does not deploy any VM in the verification process. Some other attacks may only be detected; for example, the detection of RTT manipulation. Finally, some complex attacks require more investigation to be faced. Vulnerabilities of analyzed approaches are summarized in Table III.

A. Blocking Verification

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

- Landmark Blacklisting: the CSP detects landmarks participating in the verification process and blacklists them because they are potential witnesses of the malicious CSP behavior. By preventing landmarks to collect RTT measurements, the verification process is blocked.
- Trace service blocking: the most common way to measure the RTT is to use ICMP (Internet Control Message Protocol) queries such as *Traceroute* and *Ping* queries. Under the pretext of security or performance reasons, the 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.

2) Solutions: 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 the user and the CSP agree on a list of landmarks that will be used to verify data location. In such a case, the CSP is aware of the existence of the location verification and should avoid the 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 location verification secret, is to activate multiple legitimate machines, which are authorized to access the data on the CSP, and then collect the RTT values. The third solution is to randomly select, at each training initialization time, some landmarks

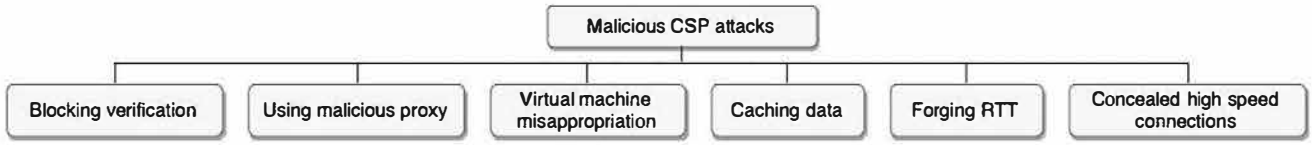

Fig. 2. Potential attacks on the location verification process

among a very large set of sites making it either infeasible or very costly for the CSP to blacklist all those sites.

- Trace service blocking: to overcome trace service blocking, the user and some landmarks may collect RTT values while accessing the data stored on the CSP. It is assumed here that the CSP cannot learn about the objectives of data accesses (i.e., when the user is really exploiting data or when the data access is just an alibi to derive RTT values).

B. Using Malicious Proxy

1) Attack principle: The CSP installs a proxy located at the data location included in the SLA, which is a valid location for the user, and stores the data at another location. Then, probing and data access queries are answered by the proxy, thus persuading the user that the data are in the right location. This attack is implemented differently depending on the interactions between the location verification process and the data access process:

- Without interaction: the verifier does not rely on the data access to verify the location. In such a case, the verification process only considers network-related metrics and the proxy only needs to reply to ICMP queries instead of the CSP. After a wave of RTT collecting what is estimated by the verification process is the 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 site, which sends the requested data to the proxy, and then the proxy sends the data to the requesting node.
- With interaction: in this case, the verification process collects the 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 later. Assuming no data caching and no special connection, the proxy may deploy attacks on the 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 the data content is relevant but no RTTs are collected. Under the assumption such a learning is feasible, when a data request is received, the proxy needs to classify it as request type *i* or *ii*. When the request is classified as type *i*, the proxy forges a random data and sends it to the user and when it is classified as type *ii*, a remote access to the site of data storage is made to receive authentic data, which are then sent to the user. Consequently, the 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 the data accesses match specific patterns (e.g., the verification process is run each Monday between 08:OO a.m. and 11:00 a.m.).

2) Solutions: Potential solutions to face this attack type are as follows:

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

C. Virtual Machine Misappropriation

1) Attack principle: In the cloud context, the data of users are stored on the CSP and also user's programs may run on virtual machines (VM) on the CSP. In such a case, both data and location verifier are on the CSP. It is worth noting that the user may choose to host, on the same CSP, totally or partially the tasks composing the verification process. However, when the CSP is malicious, it may force location verification tasks on its hosted VM not to send the valid data to derive the current data location when the CSP has changed the data location. The VM may also be moved by malicious CSPs.

2) Solution: To avoid VM misappropriation by a malicious CSP, the simplest way is not to use VM to implement the location verifier. Rather, the location verifier should be hosted by the own user's machine or by a trusted third party. However, if for any good reasons regarding user's preferences or requirements, the 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 the CSP manipulate the hosted VMs [4].

D. Forging RTT

1) Attack principle: RTT forgery is one of the basic attacks that may be used by a malicious CSP to obfuscate the landmark-based location verification approaches. When the location verifier tries to collect RTT values to derive the location of the CSP, it sends requests (*Traceroute*, *Ping* or

data access requests) to the CSP. Then, the latter delays or handles the request with a higher priority—which results in lower RTT values—its responses so that the collected RTT values either will not help the verifier to derive the current CSP location or worst the verifier derives the agreed data location (i.e., the location included in the 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 the RTT, is randomly generated. This causes the RTT to appear totally uncorrelated to the distance between the landmarks and the CSP and the verification process fails to conclude.
- Requester-location-aware RTT forgery: assuming the CSP has a certain knowledge on the verification approach and the locations and roles of landmarks, the waiting time to increase/decrease the RTT is forged depending on the origin of the query (i.e., the landmark originating the request), so that the data appears to be stored at the location included in SLA and not in the current CSP location. By forging the RTT values, the centroid of the zone where data are assumed to be is deliberately changed. Using this process, the CSP may move the centroid where it wants.

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

- Random RTT forgery: it might be reasonably assumed that the CSP would increase/decrease the RTT only for verification queries, otherwise the CSP outgoing traffic will be impacted resulting in QoS degradation. Under this observation, the verifier may detect RTT forgery by comparing the RTT values received 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 RTT computed at different landmarks. Another way to detect RTT forgery is when the zone inferred from the forged RTT measurements is too large [13].
- Requester-location-aware RTT forgery: this attack assumes the CPS is aware of landmarks' roles and positions and how the RTT values are used in the verification process to forge. To avoid this attack, a solution is to randomly select, at each training step initialization time, landmarks among a very large set of sites, which makes it either infeasible or very costly for the CSP to learn and deploy an RTT forgery for each participating landmark.

E. Caching Data

1) Attack principle: This attack may be used to obfuscate the deployment of a PDP (Proof of Data Possession) check proposed to face a malicious proxy (III.B). In caching data attack, the CSP stores most of the data at a remote location while storing in its local cache the data needed by the verification process. When a data access request is for verification, the CSP fetches the data in its local cache. Otherwise, the request is forwarded to the remote site of data storage before sending the response to the 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. It should be noticed that the data caching attack is different from the "With interaction" attack of a malicious proxy. In the latter, the 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 the data are used both by the verification process and by the conventional user's applications.

2) Solutions: There are two options to face this attack:

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

F. Concealed High-Speed Connections

1) Attack principle: The CSP is aware of the user's behavior, which shows that the user relies on RTT measured during data access to derive CSP location. The CSP chooses a site at location L, using an appropriate high-speed network, such that the delay transfer between the CSP and the chosen site is negligible compared to the variation of RTT between the CSP and the user. The CSP stores the data at the location L. Then, when the user sends data access requests to CSP, the latter reads the data from location L 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).

2) Solutions: This attack is too much elaborated as it is based on dedicated communication infrastructure. No simple solution may be suggested for such an attack without a third party to certify that the CSP does not use hidden connections. However, it can be noticed that the distance between any locations is limited by the delay induced by the 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 the necessary round-trip for the request between two sites, moving the data of 1000 km would result in 10 ms increase of RTT. RTT increase would probably be detected by the verifier.

TABLE II
ATTACKS AND COUNTERMEASURES

CSP's attack	Countermeasures
Landmark blacklisting	• Include in the SLA the list of authorized landmarks • Collect RTT by multiple machines authorized to access the data • Randomly select landmarks among a very large set
Trace service blocking	• Collect RTT values while accessing data
Malicious proxy	• Use a Proof of Data Possession Protocol • Make the CSP unable to discover the objectives of data accesses
Virtual machine misappropriation	• Do not use VM to implement the whole location verification process • Deploy on the CSP a trusted hardware to protect the VM
Random RTT forgery	• Deploy a cooperative scheme to detect suspicious RTT variation • Use the size of the derived location zone to detect RTT forgery
Requester-location-aware RTT forgery	• Randomly select from a very large set of sites, at each training step initialization time, landmarks to participate
Caching data	• Use all the data accesses to collect RTT values • Choose randomly a subset of the RTTs associated with all data accesses • Detect RTT variation between cached data and non-cached data.
Concealed high-speed connections	• Use a trusted third party to check the links used by the CSP • Use the size of the derived location zone to detect concealed high speed connections

The identified CSP's attacks and the proposed countermeasures are summarized in Table II.

IV. Vulnerabilities of Existing Approaches and Challenges

A. Vulnerabilities of Existing Approaches

In Section II, the main design features of the existing landmark-based location approaches are described. In this section, their vulnerabilities regarding the attacks identified in Section III are summarized. Table III clearly shows that almost all the proposed location verification approaches are far from being robust to thwart the attacks of malicious CSPs. Consequently, these location verification approaches have to be strengthened with appropriate countermeasures.

B. Challenges

1) Landmark Selection: Robustness and accuracy of location verification approaches rely on the collaboration of landmarks. However, the set of landmarks to be used in the location verification process has not been adequately considered in literature. At least three issues should be investigated:

- How to select—at a given cost—a set of landmarks to reach a given accuracy of the CSP location size at a given probability? And how the number of landmarks impact the security of the location verification process?
- How to randomly select landmarks to participate in each location verification wave to make the malicious CSP unable to prepare its attacks?
- How to avoid the selection of malicious landmarks and how to detect their misbehaving after being selected?

2) Statistical Learning on RTT Forgery: RTT measurements play a paramount role in landmark-based location verification approaches. RTT values are known to be variable in nature when the Internet is of concern, which makes the detection of RTT forgery a (very) complex task. One challenge is the elaboration of statistical learning models enabling the detection of RTT forgery and the performance analysis of such learning models to provide (statistical) guarantees to users regarding the CSP misbehaving detection.

3) Statistical Location Verification Guaranteeing: As mentioned previously, location verification methods are machine learning based. So their result, i.e. the derived location zone, comes with some statistical error due to the learning even when the CSP is truthful. Assuming malicious CSPs makes the error worse. Another challenge is the statistical analysis of the impact of the proposed countermeasures on the success of the location verification. In other words, given some attack scenarios and RTT variability (in normal conditions), what are the location zone and its accuracy and the level of confidence in the CSP?

4) Proof of CSP Misbehaving: Prior to the user's data transfer on the CSP, a SLA is agreed between the partners stating that the data will be stored at the location included in the SLA. In case the verification process concludes—with some probability—that either the current data location is not valid or the CSP has deployed attacks against the location verification process, how to report on this event? Then, how to reveal the location verification report proving that a CSP misbehavior has occurred? How the user can complain and prove that the CSP is malicious?

V. Conclusion

Data storage on the Cloud became one of the main services available to users via the Internet. Unfortunately, the other side of the coin of the facilities provided by cloud service providers is that users are losing some control on their data. In particular, data location on the Cloud is one of the primary concerns for Cloud users. Multiple solutions have been proposed to verify data location. This paper presents the location verification building blocks and compares the existing approaches. It also identifies the potential threats on the location verification

TABLE III
VULNERABILITIES TO ATTACKS OF ANALYZED APPROACHES

	Blocking verification	Malicious proxy	VM misappropriation	Forging RTT	Caching data	Concealed connection
Biswal2014 [10]	×	× (2)		×	×	×
Ries2011 [18]	×	× (2)		×	×	×
Fotouhi2015 [12]	×	× (2)	×	×	×	×
Jaiswal2015 [15]	× (1)	× (3)		×	×	×
Benson2011 [8]	× (1)	× (3)		×	×	×
Gondree2013 [14]	× (1)			×		×
Watson2012 [19]	× (1)			×		×
Eskandari2014 [11]			×	×	×	×

×: the proposed solution is vulnerable to the attack; an empty case means no vulnerability.
(1): Vulnerability to landmark blacklisting.
(2): Vulnerability to malicious proxy without data access.
(3): Vulnerability to malicious proxy with data access.

process in case a malicious CSP might implement attacks to jeopardize the location verification process. Malicious CSPs relocate data on low cost servers, which are not authorized by the users, and then try to make the users believe their data are on the location agreed in the SLA. We briefly describe some solutions to overcome the attacks or detect them and summarize the vulnerabilities of existing location verification approaches. We believe that integrating the proposed countermeasures in the existing approaches would make them more robust and consequently increase the trust level of users in the data storage services. Finally, we identify some challenges that require further investigation in the future. Data location verification still remains an open and exciting issue in the Cloud computing research field.

REFERENCES

[1] Canada: Personal information protection and electronic documents act. http://laws-lois.justice.gc.ca/eng/acts/P-8.6/. Accessed: 16 June 2017.

[2] 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, and repealing directive 95/46/ec (general data protection regulation). http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679. Accessed: 16 June 2017.

[3] Usa: Health insurance portability and accountability act. https://www.gpo.gov/fdsys/pkg/PLAW-104publ191/html/PLAW-104publ191.htm. Accessed: 16 June 2017.

[4] M. Achemlal, S. Gharout, and C. Gaber. Trusted platform module as an enabler for security in cloud computing. In *2011 Conference on Network and Information Systems Security*, pages 1–6, May 2011.

[5] A. Albeshri, C. Boyd, and J. G. Nieto. Geoproof: Proofs of geographic location for cloud computing environment. In *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on*, pages 506–514, June 2012.

[6] Giuseppe Ateniese, Randal Burns, Reza Curtmola, Joseph Herring, Lea Kissner, Zachary Peterson, and Dawn Song. Provable data possession at untrusted stores. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS '07, pages 598–609, New York, NY, USA, 2007. ACM.

[7] Michael Bartock, Murugiah Souppaya, Raghuram Yeluri, Uttam Shetty, James Greene, Steve Orrin, Hemma Prafullchandra, John McLeese, Jason Mills, Daniel Carayiannis, et al. Trusted geolocation in the cloud: Proof of concept implementation. *Nat. Instit. Stand. Technol. Internal Report 7904*, 2015.

[8] Karyn Benson, Rafael Dowsley, and Hovav Shacham. Do you know where your cloud files are? In *Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop*, CCSW '11, pages 73–82, New York, NY, USA, 2011. ACM.

[9] S. Betgé-Brezetz, G. B. Kamga, M. P. Dupont, and A. Guesmi. Privacy control in cloud vm file systems. In *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, volume 2, pages 276–280, Dec 2013.

[10] B. Biswal, S. Shetty, and T. Rogers. Enhanced learning classifier to locate data in cloud datacenters. In *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on*, pages 375–380, Oct 2014.

[11] M. Eskandari, A. S. D. Oliveira, and B. Crispo. Vloc: An approach to verify the physical location of a virtual machine in cloud. In *Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on*, pages 86–94, Dec 2014.

[12] M. Fotouhi, A. Anand, and R. Hasan. Plag: Practical landmark allocation for cloud geolocation. In *Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on*, pages 1103–1106, June 2015.

[13] Phillipa Gill, Yashar Ganjali, Bernard Wong, and David Lie. Dude, where's that ip?: Circumventing measurement-based ip geolocation. In *Proceedings of the 19th USENIX Conference on Security*, USENIX Security'10, pages 16–16, Berkeley, CA, USA, 2010. USENIX Association.

[14] Mark Gondree and Zachary N.J. Peterson. Geolocation of data in the cloud. In *Proceedings of the Third ACM Conference on Data and Application Security and Privacy*, CODASPY '13, pages 25–36, New York, NY, USA, 2013. ACM.

[15] C. Jaiswal and V. Kumar. Igod: Identification of geolocation of cloud datacenters. In *2015 IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops)*, pages 665–672, Oct 2015.

[16] Christoph Krauß and Volker Fusenig. *Network and System Security: 7th International Conference, NSS 2013, Madrid, Spain, June 3-4, 2013. Proceedings*, chapter Using Trusted Platform Modules for Location Assurance in Cloud Networking, pages 109–121. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[17] P. Massonet, S. Naqvi, C. Ponsard, J. Latanicki, B. Rochwerger, and M. Villari. A monitoring and audit logging architecture for data location compliance in federated cloud infrastructures. In *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on*, pages 1510–1517, May 2011.

[18] T. Ries, V. Fusenig, C. Vilbois, and T. Engel. Verification of data location in cloud networking. In *2011 Fourth IEEE International Conference on Utility and Cloud Computing*, pages 439–444, Dec 2011.

[19] Gaven J. Watson, Reihaneh Safavi-Naini, Mohsen Alimomeni, Michael E. Locasto, and Shivaramakrishnan Narayan. Lost: Location based storage. In *Proceedings of the 2012 ACM Workshop on Cloud Computing Security Workshop*, CCSW '12, pages 59–70, New York, NY, USA, 2012. ACM.

[20] T. Wüchner, S. Müller, and R. Fischer. Compliance-preserving cloud storage federation based on data-driven usage control. In *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, volume 2, pages 285–288, Dec 2013.