2014 IEEE 6th International Conference on Cloud Computing Technology and Science

VLOC: An Approach To Verify The Physical Location Of A Virtual Machine In Cloud

Mojtaba Eskandari
Fondazione Bruno Kessler, Trento, Italy and DISI, University of Trento, Italy
Email: eskandari@fbk.eu

Anderson Santana de Oliveira
SAP Labs, France
Email: anderson.santana.de.oliveira@sap.com

Bruno Crispo
DISI, University of Trento, Italy
Email: bruno.crispo@unitn.it

***Abstract*—The geolocation of data stored and being processed in cloud is an important issue for many organisations due to obligations that require sensitive data to reside or be processed in particular countries. In this paper we introduce an approach, named VLOC, to verify the physical location of a virtual machine on which the customer applications and data are stored. VLOC is implemented as a software which is able to estimate the geolocation of itself and notify the corresponding user if the location is unauthorised. VLOC uses a number of arbitrary web-servers as external landmarks for localisation and employs network latency measurement for distance estimation. Due to the fluctuation in the network latency, VLOC employs a machine learning technique in order to adapt itself to various network latency tolerance. Different from most of geolocation estimation approaches, VLOC is installed inside the target host (inside the cloud). VLOC does not require special hardware nor a network of trusted landmarks. The experimental results shows the accuracy of VLOC is higher than other existing approaches.**

***Keywords*-geolocation; cloud security; privacy;**

I. Introduction

According to the National Institute of Standards and Technology (NIST), one of the essential characteristics of cloud computing is resource pooling which allows cloud service providers (CSPs) to serve multiple consumers using a multi-tenant model by dynamically assigning resources on consumers' demand [1]. Cloud service provisioning is independent of the location of the provider, as the services are consumed over the Internet. CSPs wish to be free to relocate data for load balancing purposes in order to reduce the maintenance cost. However, knowing and controlling the physical location of data for storage and processing purposes could be very important for organization using Cloud in some particular scenarios dealing with compliance [2]. A piece of sensitive data may be transferred amongst various data centres situated in different geographical locations, and consequently there might be violations in data privacy as there are various regulations for privacy protection in different countries. Furthermore, there are a number of specific obligations about storing and processing sensitive information in a set of particular geographical areas such as European Union Data Protection Directive [3].

Thus, cloud users would benefit from a service that could verify the physical location of their data. There are a number of approaches for finding the physical location of a piece of data or a host. Generally, they take advantage of network metrics such as round trip time delay for a transmitted message between two identical hosts and then calculate the distance or the physical location of one of hosts based on the measured latency from the other ones. The main drawback of this approach is dynamicity of the internet. As the network load changes frequently in time, it is not possible to find a constant correlation between network latency and physical distance. In addition, there are other factors which impose delay on a transmission such as authentication mechanisms, network delays, proxying, caching, and so on. Therefore, an adaptive approach is required to deal with the dynamic environment of the Internet.

In this paper we introduce a geolocation approach, named *VLOC* (a Verifier for physical LOCation of a virtual machine), which is able to verify the physical location of a virtual machine by taking advantage of nearby randomly chosen web-servers. Since VLOC does not rely on a network of fixed landmarks, its implementation is easier and maintenance cost lower than other proposed solutions. VLOC is implemented as a software component which needs to be installed and initialised on a virtual machine.

The rest of the paper is organised as follows. The following section describes the system model then Section III explains VLOC in detail. Section IV evaluates VLOC and discusses the experimental results. Section V outlines the related approaches and Section VI concludes the paper.

II. System Model

- The **list of websites**; A database of websites addresses. For instance, these addresses can be collected from Alexa [4].
- The **VLOC tool**; it is a software component installed on a virtual machine to verify its physical location. VLOC includes a **Data Collector** component which collects the required geolocational information for every website. The **IP location service** provides geolocational information of the web-server of a particular website.

978-1-4799-4093-6/14 $31.00 © 2014 IEEE
DOI 10.1109/CloudCom.2014.47

The **Round-Trip-Time (RTT) measurement module**, which measures the network latency between current virtual machine and a target web-server by sending multiple HTTP requests to the website hosted on that web-server. The number of HTTP requests can be specified through a parameter passed to this module. Finally, the average value of round trip time of the successful requests is returned as a result.

- The **target virtual machine**; This is the virtual machine that needs to be securely geo-located and on which the VLOC tool is installed. The virtual machine holds data as well web services users want to run on the cloud.
- **Current host**; it is the physical server on which the virtual machine is running.
- The **distance estimation function** which maps each RTT value to a distance between the pair of associated hosts. This function is a polynomial function and its coefficients are variable and updated during the initialisation of the VLOC tool. The value of coefficients are calculated based on collected data and their corresponding measured RTT values. Therefore, the function is able to estimate the distance between two identical hosts based on former observations.
- The **learning module** calculates the coefficients of the distance estimation function by finding the correlation between measured RTT values of two identical hosts and their associated distance. This module attempts to find the best function approximation which represents the collected data.
- The **triangulation technique**; this is a technique used to specify the physical location of a point by having the latitude and longitude coordinates of at least three nearby points. This technique is used, in our model, to estimate the physical location of the current host based on three nearby web-servers.

III. VLOC

The user can install the VLOC on her virtual machine and once the tool gets initialised, it notifies the user the physical location of the virtual machine. VLOC does not need a dedicated physical device nor a network of pre-arranged landmarks. The main requirement of VLOC is the availability of an online IP geolocation service like [5], [6], to get a list of websites like *Alexa 1-million* [4] and the current geolocation of the virtual machine. First, the tool chooses a, configurable, number of random websites (agreed with the CSP at the moment user buys its cloud hosting service) and then starts to collect geolocation information about them. Then, it can verify at any moment, the geolocation of the virtual machine.

Figure 1. Triangulation procedure to find the physical location of a host by knowing the physical locations and distances from other hosts [8].

A. Recognising the Physical Location of A Virtual Machine

To find the physical location of a server on which a virtual machine is installed, a feasible solution is to take advantage of some quality of service metrics used in networks. In [7], multiple trusted landmarks with known physical locations are used. The distance between a landmark and a data centre is obtained by sending specific messages and measuring transfer delay with a an error below a chosen threeshold. At least three landmarks are required to achieve the required accuracy in triangulation procedure. Figure 1 shows an example of the localisation mechanism. In this example, the virtual machine installed on a host situated in *Trento* (shown by the blue marker) is asked to verify if its physical location has been modified without authorisation. The virtual machine sends HTTP requests to three hosts and measures the round trip time delay. Then, based on measured delays, distance from every host is estimated and by utilising a triangulation technique, the physical location of the host on which the virtual machine is installed will be computed.

As mentioned before, VLOC needs an initialisation which consists of three phases. The first phase is to collect geolocational information of the given list of websites. Algorithm 1 shows the procedure of data collection. This algorithm takes a list of websites, an online IP geolocation service, and the location of the current host (the virtual machine) and then finds the physical location of web-server of each website and calculates the distance between the web-server and the current host and stores them into a database. The second phase of initialisation, which is depicted in Algorithm 2, is measuring the round trip time (RTT) delays of websites. This algorithm takes the list of websites, a range of operation which signifies the radius of a circle showing a geographical zone, and a confidence factor and then it measures the RTT value of an HTTP request for

every website. As using long distances increases the error rate of distance estimation, our approach limits its range of operation to the nearby websites and in Algorithm 2 the range of operation refers to choosing websites situated in range of R KMs. Due to probability of failure in the requests and the delay of packet routing imposed on some requests, this algorithm takes a parameter named confidence factor C which repeats the HTTP transmission operation C times for each website and finally the average of successful HTTP requests is used. Since after initialisation phase the physical location of the current host needs to be verified and the only trustworthy entity is network delay measurement, it is required to provide a function which maps an RTT value to the corresponding distance. Having distance from at least three hosts enables the current host to calculate its physical location by making use of a triangulation technique. Therefore, the last phase of initialisation is to prepare a function being able to estimate the physical distance between the current host and an arbitrary host.

Input: L: list of websites; IPG: reference of IP geolocation service; H: current host information;
Output: L': List of websites with their collected geolocation information;

L' = ***new*** List();
for *(w in L)* **do**
    g = IPG.*getInfo*(w);
    d = $distance(H, g)$;
    $r = \{w, g, d\}$;
    add r to L';
end
return L';

Algorithm 1: The data collection algorithm.

The distance estimator function, in VLOC, employs a distance bounding protocol in order to calculate distance between two geographical points based on their measured RTT value. The following equation shows a simple distance calculator:

$$f(x) = a.x \tag{1}$$

where x is the given RTT value and a is a coefficient that converts the value of a round trip time delay to its corresponding distance. Unfortunately, due to dynamicity of packet transmission in the Internet, it is not possible to consider a constant coefficient for the distance estimation function. Moreover, the hierarchical architecture of cloud does not allow the protocol to work properly as in order to transmit and process a request, the request needs to pass through various service layers. In addition, each layer imposes an extra delay on the process and the number of participated service layers is vary per different types of requests. Therefore, in order to estimate the transmission

Input: L': List of websites with their geolocation information;
R: Range of operation;
C: Confidence factor;
Output: L'': List of chosen websites with measured RTT;

L'' = ***new*** List();
for *(r in L')* **do**
    if *($r_d < R$)* **then**
        for $i = 1$ *to* C **do**
            Send an HTTP request to r_w;
            t_{start} = Now();
            Wait for respond from r_w;
            res = The received response;
            t_{end} = Now();
            if *(res was successful)* **then**
                $\Delta t_i = t_{end} - t_{start}$;
            end
        end
        $rtt = \langle \Delta t_{1...C} \rangle$; *// Average*
        $rec = \{w, rtt\}$;
        add rec to L'';
    end
end
return L'';

Algorithm 2: Measuring and collecting round trip time (RTT) latencies of the nearby websites.

latency of a request, it is not possible to consider a global constant coefficient for the distance calculation function. Thus, a technique is required being able to adapt itself with various circumstances and handle different delays in the distance calculation procedure. VLOC uses the following equation:

$$f(x) = \sum_{i=0}^{n} a_i x^i \tag{2}$$

in this equation the coefficients are variable and they are updated according to the observation performed by Algorithm 2. This algorithm chooses a random subset of websites from the list and measures the RTT values for them and then updates the coefficients. In order to provide an accurate observation, a sufficient number of websites must be used (e.g. at least 500 websites). Therefore, the algorithm is able to cover the regular turbulences happening in the network as depicted in Figure 2, the coefficients do not face abrupt changes; they stay in a limited range.

Figure 3 illustrates an example of the distance estimation function. As this figure shows, each item (*i.e.* website) has an RTT value and a corresponding distance from the current host. The coefficients of the function are obtained by applying a machine learning technique (*i.e.* Polynomial Regression [9]) on the collected data. The function shown

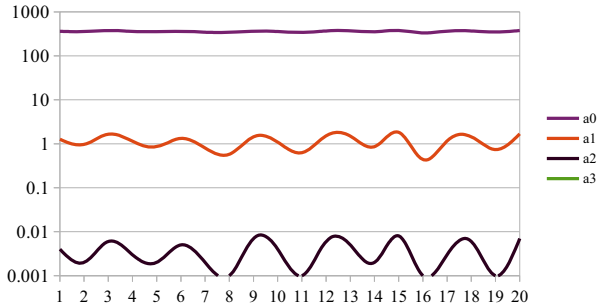


Figure 2. An observation on the changes of the coefficients of Equation 2 during the update process captured 20 times. These results show that choosing a random subset of websites for each update, does not lead to very different coefficients.

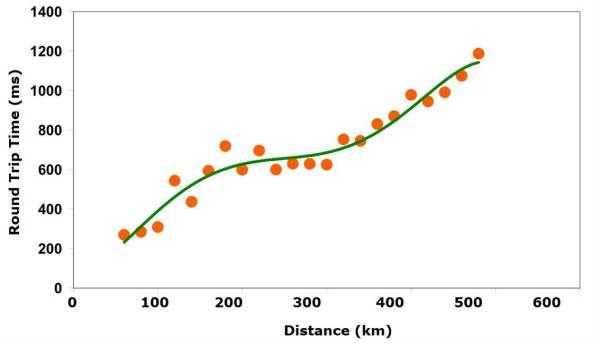


Figure 3. A sample of collected RTT values versus distances and the trained function representing the distance estimation procedure.

in this figure is an example of trained function which is able to perform distance estimation for further RTT values. Therefore, employing this technique enables us to handle the dynamicity of the Internet environment.

An interesting question might arise here, since the environment of the cloud computing is changing in time the distance estimation based on one time observation might not be accurate enough. In fact, this is likely to be true because network and host conditions may be different at the time of observation and at the time of estimation. Therefore, there must be a short time gap between observation and estimation. However, the size of this gap depends to the fluctuation of latency of the network. In order to maintain the accuracy above an accepted level, the observation needs to be performed periodically. In fact, while the estimation function is being used, the coefficients can be updated frequently and provide better accuracy based on the most recent observations. In order to utilise such a technique, we need to perform the observation procedure in every predefined time slot; and then update the coefficients based on them. The entire process is depicted in Figure 4.

Once the initialisation phase is finished, the distance estimation function is ready to be used. In order to verify

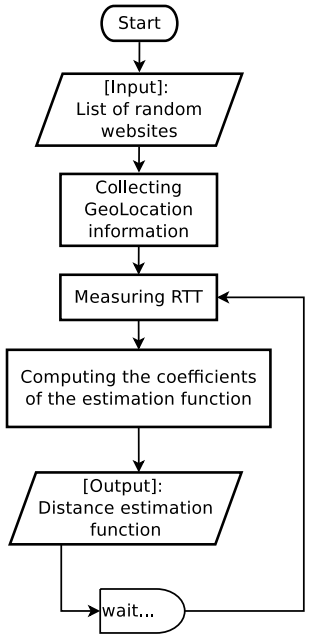


Figure 4. The initialisation process. The process of frequently updating the coefficients of the distance estimation function is illustrated in this figure.

the location of current host (virtual machine), at least three nearby websites get selected and then by making use of the mechanism employed in Algorithm 2, the RTT value for those hosts are measured. In the next stage, the distances between current host and the selected websites are estimated by utilising the distance estimation function. Finally, as illustrated in Figure 1, the geolocation of current host is obtained.

The presented approach can be integrated into various techniques and schema in order to be used in various application. For instance, it can be integrated into a proof of retrievability service (PoR) such as [2], [10], [11], [12], [7] or into a data transfer monitoring framework such as the one introduced in [13]. Moreover, it can be used as a notification service on transferring a specific piece of data to an unauthorised physical location.

B. Security Considerations

Alternative approaches to ours use a network of pre-arranged landmarks, situated outside of the target host, as verifiers. They send challenge messages to the host and measure the RTT values and finally estimate the physical location of the host. All these approaches however, require an external network of landmarks.

In contrast, our approach puts the verifier inside the target host, removing in this way the requirement of an organised network of landmarks. In VLOC, the physical location is estimated by sending message from the target hosts to existing websites, rather than from some specified landmarks to the target host.

This however, opens security issues that VLOC needs to address. Since VLOC is in the virtual machine hosted on the cloud, a mistrusted cloud provider could intercept and manipulate all communications between VLOC and the websites. Encrypting the messages is not a solution, since the encrypting key would reside on the cloud and can be extracted from the RAM by the cloud provider. Our solution to this problem is to obfuscate the communications VLOC performs for the purpose of estimating the distance within the regular traffic of applications stored on the virtual machine. The reasoning behind this choice is that the cloud provider could not easily filter out or block these messages and a full packet inspection would be required. Since in our threat model the cloud provider moves data only for the purpose of saving money, breaking our system would simply require more effort than what gained by moving the data.

Therefore, VLOC does not use fixed landmarks, easy to blacklist, but rather randomly chosen websites as external landmarks. In order to measure required RTT values, we use normal HTTP requests which would be difficult to block without affecting other applications. The cloud provider sees the virtual machine sending an HTTP request to some websites like what many applications do for REST requests or SOAP ones.

An other possible point of attack is the list of websites VLOC will query. Rather than embedding a fixed list in VLOC software, the user can configure online and dynamically at her will the address of the IP-location service VLOC uses to gather list and location of the candidate websites. The size of the list and the number of selected website can be configured dynamically as well.

C. Limitations

Although VLOC is promising in a practical environment, there are a number of limitations need to be considered. One of the limitations is related to detection of network latency changes (*i.e.*, due to network disruption). This may have an impact on the accuracy of the estimated location. While an adaptive monitoring module capable of such detection is under development, at the moment VLOC adopts the strategy of periodically repeating the measurements. The frequency of such confirmation is a parameter can be configured dynamically.

Furthermore, there are two other parameters in VLOC need to be tuned in order to achieve the best accuracy. Those parameters which are the range of operation, R, and the confidence factor, C, impact on the amount of noise in the measured latencies and consequently on the accuracy of the geolocation estimation procedure. The confidence factor plays a crucial role in the measurement phase as it attempts to handle the fluctuation of the network while the duty of the range of operation is to filter out far web-servers. Since long distances overwhelm the impact of short distances in training, they reduce the accuracy of geolocation estimation as it is demonstrated in Figure 8.

As the cloud provider is considered as an adversary, it can perform some operations to reduce the accuracy of VLOC. For instance, it can inject packet delays on all outgoing and incoming traffic. These delays could be recorded, or they could be randomly generated. At the time of injecting these delays, VLOC faces a slight reduction in accuracy, however, since VLOC observes the environment frequently and adapts itself with the network turbulences, it can get adapted to the such a situation. The cloud provider is agnostic to the purpose of outgoing and incoming packets as VLOC packets are exactly regular HTTP requests. Furthermore, if the cloud provider performs such an operation, it impacts the quality of the service which is an important key in cloud business.

IV. Empirical Setup

In order to evaluate VLOC, we developed it in form of a web-based tool in PHP/MySQL which collects the data and executes the training and accuracy measurements. The target host is a computer in Trento, Italy and the goal is to estimate the geolocation of this computer.

This section explains the data collection process and describes the data used. It also describes the evaluation measures, the experimental results and their analysis.

A. Data Collection

As mentioned before, the initialisation phase needs to collect geolocational information and measure RTT latencies of a number of randomly chosen websites. In order to do so, we used *Alexa 1-million* [4] list from which the geolocational information of $188,644$ websites were collected. We have used *IPaddressAPI.com* [5] as the IP-location service. This operation is performed by Algorithm 1. After collecting such information, we selected $38,892$ websites which were geographically located in the vicinity of 1000 KM radius. We measured the RTT values of these website by employing the Algorithm 2. Figure 5 illustrates the number of these websites (used as landmarks) in various ranges. The confidence factor for this measurement was set to 10, which means for every website 10 HTTP requests were sent and the average of successful ones was stored as the corresponding RTT value. The HTTP requests used in the experiments were sent through a PHP function named *fsockopen* [14] which initiates a socket connection to a resource in network. We used this function to open a connection to a given website address on port 80 referring to the port of http protocol. Since the application only opens a socket and does not download any web-page, it acts resembling a ping request over HTTP protocol.

B. Evaluation Measure

Since the main purpose of VLOC is to verify the location of a virtual machine, the evaluation measure must be able

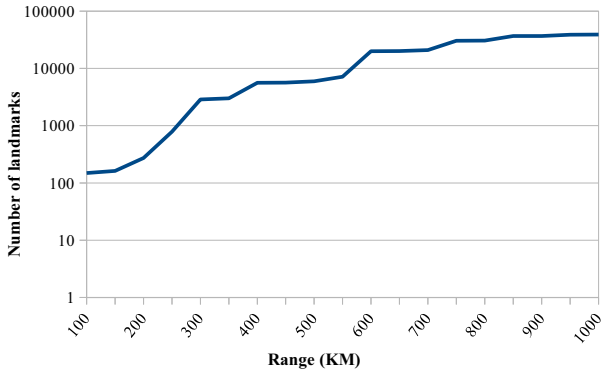


Figure 5. The number of used landmarks (websites) per various ranges. Each range refers the maximum distance between the landmarks and the current host.

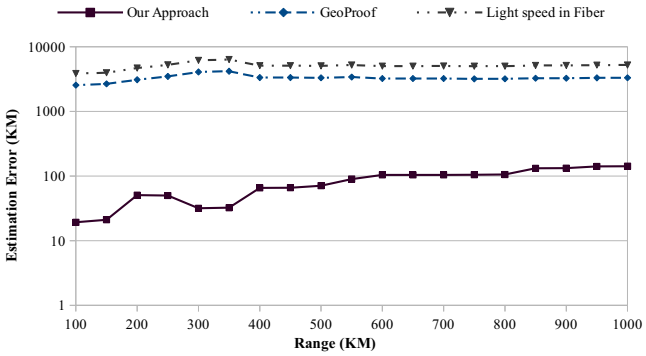


Figure 6. A comparison of various distance estimations done by VLOC and its rivals. These results show the average of estimation error in various ranges.

to verify the distance between the actual physical location of the machine and its estimated location. We used error of average distance estimation defined in the Equation 3 for accuracy evaluation.

$$E_{avg} = \frac{1}{N}\sum_{i=1}^{N} \|p_e^{(i)} - p_o^{(i)}\| \tag{3}$$

where N is the number of data instances participating in test phase, $p_e^{(i)}$ denotes the estimated physical location for i^{th} website in the list and $p_o^{(i)}$ is the observed geolocation (the real physical location) for that website. Finally, E_{avg} refers to the calculated average error in KM.

In order to provide a comprehensive evaluation, we evaluated the approach with different websites which is achieved by applying random combination of measured RTT values. We used cross validation technique [15] to perform such a combination by using 10 fold 5 times setting which works as follows. First, the collected data is divided into 10 parts called folds. Out of these 10 parts, 9 are used to construct the estimation function. The remaining 1 fold is used to evaluate the constructed function. Then the data items get shuffled and the same division and evaluation is repeated for 5 times.

C. Accuracy

In this section the accuracy of location estimation is discussed. In order to estimate the physical location of a server, first we need to estimate the distances of at least three nearby hosts. The accuracy of distance estimation makes a major impact on the accuracy of physical location estimation (i.e. triangulation procedure). We also compare the accuracy of VLOC with other distance measurement techniques which are GeoProof [2] and distance calculation with the speed of light.

GeoProof uses the following equation for its distance measurement:

$$f(x) = \frac{1}{2}x\frac{4}{9}s10^{-6} \tag{4}$$

where x is the given RTT value, $\frac{4}{9}s$ is the measured speed of transferring data over the Internet while s is the speed of light. This function $f(x)$ takes x in milliseconds and computes the distance in KM. Since there is no accuracy evaluation experimental results provided by GeoProof in their paper, in order to compare its accuracy with the accuracy of VLOC, we applied GeoProof distance estimation formula on our measured RTT latencies.

The other experiment is done by using speed of light in fiber as the calculation parameter which would be as follow:

$$f(x) = \frac{1}{2}x(0.66)s10^{-6} \tag{5}$$

As the speed of light in fiber is 66 % of the speed of light in vacuum, it can be used as measure for distance calculation. However, this measurement can be solely used in a high speed network with neither routers or other kind of nodes in the middle. We consider it as a theoretical baseline. VLOC provides a more realistic correlation between distance and observed network latency over the Internet. VLOC builds a model representing such a correlation and uses it for distance calculation. Since the model is build based on observation of network latency regardless the type of network environment, it is able to estimate the distance between two hosts based on their message transmission latency. In addition, building a specific model for the current network and updating the model based on the changes in the latency of the network enable it to handle the fluctuation of the transmission latency. This is achieved by tuning the coefficients of the estimation function, introduced in Equation 2, with the measured RTT values. Figure 6 shows the impact of adaptive approach on the accuracy of distance estimation and compares it with non-adaptive approaches.

Once the estimation function is constructed, VLOC will be able to verify the physical location of the current machine.

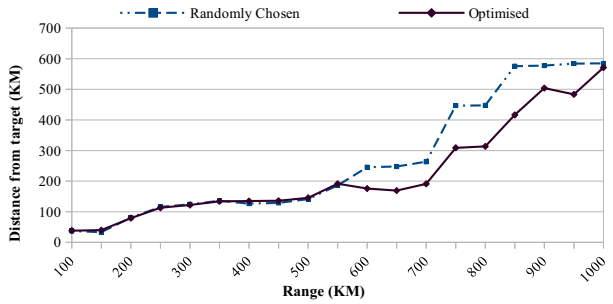


Figure 8. Estimation error in localisation per various ranges. This figure depicts the results in two landmark selection styles which are optimised and random selection.

In this stage we provide the accuracy of geolocation estimation. The location of landmarks makes a significant impact on the accuracy of localisation in triangulation technique. Figure 7 illustrates this impact, which shows the current server needs to be surrounded by the chosen landmarks and using randomly chosen landmarks. Randomly chosen landmarks do not guarantee the best accuracy. According to this fact, we performed the experiment of localisation estimation in two fashions including randomly chosen landmarks and optimised ones. The results of these experiments are depicted in Figure 8. These results reveal that as the range of operation increases, the optimised chosen landmarks outperform the randomly chosen landmarks.

According to the results shown in Figure 8, the best result for geolocation estimation is obtained in range of $150\ KM$ in which 162 landmarks are participating. As the range of the operation grows, the accuracy of the location estimation falls down. In order to utilise triangulation technique in larger ranges, we need to draw larger circles which increases the risk of estimation error. Thus, the landmarks situated in nearby are the best for our purpose.

In VLOC, various factors impact on the accuracy represented by the following statement:

$$Acc \propto \frac{P \times C}{F} - \| \frac{d}{dR} f(R) \| \tag{6}$$

where Acc is the accuracy, P is the frequency of performing RTT latency measurement in order to keep an updated observation of the network latency, C is the confidence factor used in Algorithm 2, F refers to the network fluctuation which is obtained by calculating the latency differences of a number of HTTP requests transmitted between two identical hosts. $f(R)$ is a function representing the changes of accuracy based on changes of range of operation, R. Small values of R (*i.e.* less than 100 KM in our experiments) do not yield an accurate result because the number of landmarks in small ranges are not sufficient. On the other hand, as Figure 8 shows, there is an optimum point for R and increasing this value after that point makes a negative impact on the accuracy. Therefore, in this statement, the derivative of such a function is used.

V. Related Work

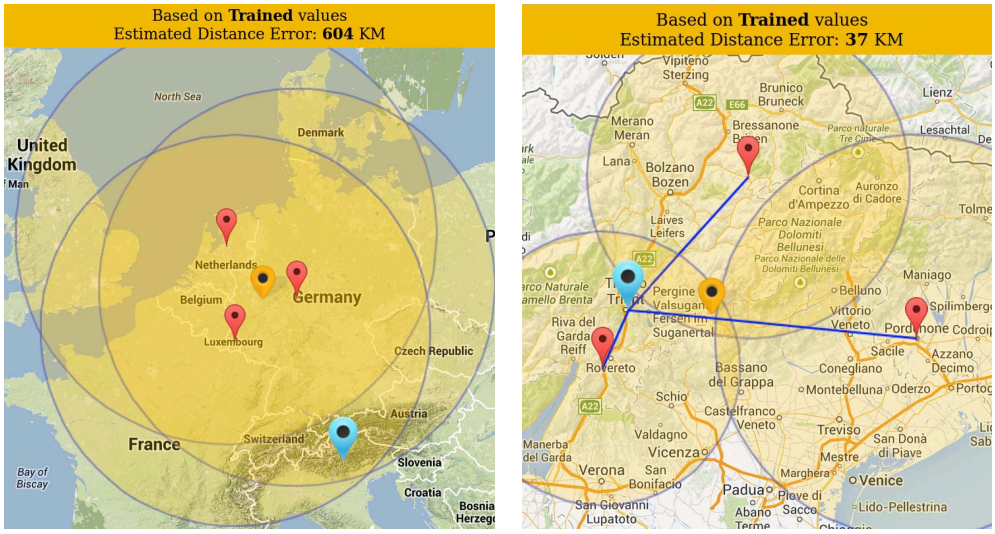
Peterson et al. in [7] introduced the idea of combining the concept of Internet geolocation with Proof of Retrievability (PoR) for data localisation. *GeoProof* is an implementation of such an idea [2]. It uses a tamper-proof physical component installed in the local network of cloud servers. As this component is GPS [1] enabled, it is able to recognise its own location. In addition, GeoProof employs a PoR protocol [10] by which it challenges the storage servers. The information gathered from the PoR protocol and the physical component enable it to verify the location of a piece of data.

The major drawback of GeoProof is the requirement of a tamper-proof and GPS enabled device situated inside the local network of each data centre. Cloud providers may hesitate to adopt such solutions as it may leak sensitive information. In [16] GeoProof is enhanced by reducing the required computational overhead and improving its accuracy, but the mentioned drawback remains unresolved.

As distance bounding protocols such as [17], [18], [7], [2] use network latency for distance calculation, they are quite time critical. Therefore, network fluctuation significantly decreases their accuracy. Network latencies can be imposed by network equipments and servers. Such latencies can not be distinguished from message transmission latency. Hence, distance bounding protocols suffer from lack of accuracy in dynamic environments such as Internet. Gondree and Peterson proposed a schema to tackle such problem by employing a latency function built based on the current network traffic observation [19]. In their schema, there are a number of landmarks which observe the network traffic by transmitting a number of messages amongst themselves and then build a model based on that. The main disadvantage of this approach is the requirement of a dedicated network of landmarks which is quite costly. Moreover, in the model building phase the landmarks send messages amongst themselves in order to find a baseline for the Internet delay which does not quite represent the real environment. In fact, this scenario does not consider the latencies imposed by cloud mediation services such as authentication, decryption, etc. Therefore, the observation has an inherent error which influences the distance estimation.

DLAS provides a data localisation assurance service based on cryptographic foundations that allows cloud users to select the preference regarding data location [20]. In order to provide such service, DLAS uses a Zero Knowledge System (ZKS) protocol to maintain secrets and verify them as mentioned in the Service Level Agreement (SLA) between parties. In DLAS, the CSP (called enterprise in that paper) is trusted and uses an external cloud storage service and

[1]Global Positioning System



(a) An example of extremely bad chosen landmarks.

(b) An example of desirable chosen landmarks.

Figure 7. Two observations of randomly chosen landmarks which can be perfect or can give very different location estimation. The light red markers show the locations of the selected landmarks, the blue marker is the current host (Trento), and the yellow marker points to the estimated physical location of current host.

guarantees not to move user's data according to her location preferences. The storage provider (SP) prepares a list of all data centres with their physical locations and informs the CSP once a piece of data is moved. Employing ZKS protocols enables the CSP to verify the region of a particular data centre and prevents the CSP from violation of the data location preferences policies. Since DLAS does not use any external resource for geolocation and relies on logical characteristics of data centres, it is vulnerable to be bypassed by virtualisation. A copy of network topology of all data centres (*i.e.* empty virtual machines and settings) can be stored on each data centre and a piece of data can be moved amongst them without awareness of DLAS. Our approach, VLOC, does not suffer from this kind of attack.

Massonet et al. introduced a system which monitors data transfers by making collaboration between cloud infrastructure provider and the service provider (i.e. user) [21]. In this system, data controller (*i.e.* tenant or cloud customer) is able to specify required locations for a piece of data allowing to be processed and the system prevents moving data to unauthorised locations. However, its major drawback is providing such a monitoring service only at infrastructure level. Therefore, it does not cover data items with finer granularities. This drawback is resolved by another work [13]. It introduces a vast monitoring framework being able to collect evidences about data transfers in various service levels. Basically this framework employs a dedicated monitor for each of service layers including $SaaS$, $PaaS$, and $IaaS$. Each monitor tracks the API calls related to data transferring and stores required logs. Furthermore, in order to track the movements of a piece of data in various layers, this framework keeps a map amongst different granularities for the data. This framework is promising; however, there is an assumption which says the CSP wishes to demonstrate compliance; therefore, it does not move user's data without authorisation. This assumption is quite reasonable as there are many ways to make a copy of data without having authorisation. However, restricting the known ways of copying and transferring data and employing a geolocation technique mitigate the risk of illegal data transferring. Due to this assumption, CSP provides a list of all data centres with their physical locations. In our attack model, we assume that the CSP is not trustworthy as its goal is to minimize maintenance costs by moving resources to less expensive data centres.

VI. Conclusions

This paper presents an approach, named VLOC, for verifying the physical location of a virtual machine without using a network of fixed external landmarks nor a GPS enabled device. VLOC is implemented as a software which able to estimate the physical location of itself and notify the corresponding user if the location is unauthorised. It allows a user to install it on a virtual machine and after initialisation it will be ready to be practically used.

VLOC works inside of the target host (inside of the cloud) and does not rely a network of fixed external landmarks; therefore, the implementation cost is quite negligible. All

a user needs to do is to install it as a tool on his/her virtual machine and then initialise it. However providing a geolocation service by using a tool installed inside the cloud while the cloud provider is the major adversary brings an important security issue. Since cloud provider has control over the infrastructure, platform, and the network, he is able to modify the real measurements with fake information. Our strategy against such an attack is to use random websites as external landmarks and obfuscate our messages into a regular protocol such as HTTP. In this scenario, it is significantly costly for the cloud provider to filter the network traffic and modify the information.

The experimental results demonstrate that VLOC is accurate enough for being used in practice. Moreover, it can be integrated into a monitoring framework in order to track a piece of data or into a policy enforcement engine as a policy information point in XACML architecture [22].

ACKNOWLEDGMENT

This work has been partly supported by the EU under grant 317387 SECENTIS (FP7-PEOPLE-2012-ITN).

REFERENCES

[1] P. Mell and T. Grance, "The nist definition of cloud computing," 2011.

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

[3] P. D. Hert and V. Papakonstantinou, "The proposed data protection regulation replacing directive 95/46/ec: A sound system for the protection of individuals," *Computer Law & Security Review*, vol. 28, no. 2, pp. 130–142, 2012.

[4] Alexa. (2014) Alexa.com. [Online]. Available: http://www.alexa.com

[5] IPaddressAPI.com. (2014) Ipaddressapi.com. [Online]. Available: http://www.ipaddressapi.com/

[6] MaxMind.Inc. (2014) Freegeoip.net. [Online]. Available: http://freegeoip.net/

[7] Z. N. J. Peterson, M. Gondree, and R. Beverly, "A position paper on data sovereignty: The importance of geolocating data in the cloud," in *Proceedings of the 3rd USENIX Conference on Hot Topics in Cloud Computing*, ser. HotCloud'11, 2011.

[8] Wikipedia. (2014) Trilateration. [Online]. Available: http://en.wikipedia.org/wiki/Trilateration

[9] Wikipedia. (2014) Polynomial regression. [Online]. Available: http://en.wikipedia.org/wiki/Polynomial_regression

[10] A. Juels and B. S. Kaliski, Jr., "Pors: Proofs of retrievability for large files," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, ser. CCS '07. New York, NY, USA: ACM, 2007, pp. 584–597.

[11] S. Kamara and K. Lauter, "Cryptographic cloud storage," in *Financial Cryptography and Data Security*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, vol. 6054, pp. 136–149.

[12] A. Noman and C. Adams, "Providing a data location assurance service for cloud storage environments," *J. Mob. Multimed.*, vol. 8, no. 4, pp. 265–286, 2012.

[13] A. De Oliveira, J. Sendor, A. Garaga, and K. Jenatton, "Monitoring personal data transfers in the cloud," in *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, vol. 1, 2013, pp. 347–354.

[14] PHP. (2013) fsockopen function. [Online]. Available: http://php.net/manual/en/function.fsockopen.php

[15] T. Hastie, R. Tibshirani, and J. Friedman, "Model assessment and selection," in *The Elements of Statistical Learning*, ser. Springer Series in Statistics. Springer New York, 2009, pp. 219–259.

[16] A. Albeshri, C. Boyd, and J. Nieto, "Enhanced geoproof: improved geographic assurance for data in the cloud," *International Journal of Information Security*, vol. 13, no. 2, pp. 191–198, 2014.

[17] J. Reid, J. M. G. Nieto, T. Tang, and B. Senadji, "Detecting relay attacks with timing-based protocols," in *Proceedings of the 2Nd ACM Symposium on Information, Computer and Communications Security*, ser. ASIACCS '07. New York, NY, USA: ACM, 2007, pp. 204–213.

[18] G. Hancke and M. Kuhn, "An rfid distance bounding protocol," in *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*, 2005, pp. 67–73.

[19] M. Gondree and Z. N. Peterson, "Geolocation of data in the cloud," in *Proceedings of the Third ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '13. New York, NY, USA: ACM, 2013, pp. 25–36.

[20] A. Noman and C. Adams, "Dlas: Data location assurance service for cloud computing environments," in *Privacy, Security and Trust (PST), 2012 Tenth Annual International Conference on*, 2012, pp. 225–228.

[21] 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*, 2011, pp. 1510–1517.

[22] OASIS-Standard, "extensible access control markup language (xacml) version 2.0," 2005.