

Verification of Data Location in Cloud Networking

Thorsten Ries*, Volker Fussenig†, Christian Vilbois* and Thomas Engel*

*Interdisciplinary Centre of Security, Reliability and Trust, University of Luxembourg, Luxembourg

†Fraunhofer Research Institution AISEC Network Security & Early Warning Systems (NES), Munich, Germany
 {thorsten.ries, thomas.engel}@uni.lu, volker.fussenig@aisec.fraunhofer.de, christian.vilbois.001@student.uni.lu

Abstract—Cloud computing aims to provide services and resources on a pay-as-you-use basis with additional possibilities for efficient adaptation of the required resources to the actual needs. Cloud networking extends this approach by providing more flexibility in the placement, movement, and interconnection of these virtual resources. Depending on the use, customers however require the data to be located under a certain jurisdiction. To ensure this without the need of trusting the cloud operator, we propose a geolocation approach based on network coordinate systems and evaluate the accuracy of three prevalent systems. Even if the cloud operator uses supplemental measures like traffic relaying to hide the resource location, a high probability of location disclosure is achieved by the means of supervised classification algorithms.

I. INTRODUCTION

Under the name of cloud computing companies offer different resources, e.g., processing, storage, and networking, to customers. The abstraction level of these differs from Infrastructure-as-a-Service (IaaS), which means that the cloud operator offers pure hardware functionalities like hard disk space or resources for a virtual machine image (e.g., Amazon's EC2 [1]), over Platform-as-a-Service (PaaS), where the cloud operator offers a platform on which applications of customers can be run (e.g., Google's App Engine [2]), to Software-as-a-Service (SaaS), where the cloud operator offers applications to the customer (e.g., Google Docs [3]).

The concept of cloud networking extends cloud computing in a way that it allows more flexibility in the placement, movement, and interconnection of virtual resources. Optimally, cloud networking crosses operator's border in a way that virtual resources can be connected automatically and can be moved automatically from one operator's side to another. With such flexibility new possibilities for optimisation arise, i.e., virtual resources can be placed in a way such that network load, the overall cost for the customer, or the latency of access is minimised.

In today's cloud infrastructures the customer often does not exactly know in which data center a virtual resource is placed and if the operator gives information on the placement the customer cannot be sure if this information is correct. Even if this information is initially correct the cloud operator might move the virtual resource to another data center.

In both scenarios, cloud computing and cloud networking, customers want to protect their data by establishing security policies that need to be followed by the provider. To a certain extend, these can be verified by manually checking the policy properties. However, this process can result in disproportionate

large efforts, if, as already mentioned the location of data needs to be verified. Therefore, additional measures are needed. In this paper we present such a mechanism for verifying the location of virtual resources without the need of trusting the cloud operator.

The location of data as part of a security policy is a result of different legislative domains where virtual resources can be placed and also the (often subjective) trust level of different countries. For example American companies don't want to process critical data outside of America, whereas companies from European countries probably may not want to move critical data to the US because of the patriot act [4] that allows US governmental institutions to access all electronic data.

Current approaches for geolocation can be classified to measurement based and semantic based geolocation approaches, both often used as basis of public databases [5], [6]. However, both approaches show certain disadvantages and do not really work if the device with the target IP-address relays the packets to the desired destination. In this particular case the cloud operator might move the virtual resource to another data center and relaying all packets to and from the virtual resource via the original IP-address. In this example both geolocating techniques are not able to detect the new placement of the virtual resource.

To overcome this shortcoming we propose the geolocating techniques to be based on virtual network coordinate systems (VCS). Through *round trip time* (RTT) measurements between different locations in the Internet to the virtual resource, the VCS allows an estimation of geographic locations. We consider the virtual resource to be a Virtual Machine, which is able to respond to the RTT requests and assume two cloud topologies: first, the cloud resource to be connected directly to the Internet and second, the cloud resource being placed behind a relay node, providing the possibility of transparent changes for the cloud provider. Based on these settings, we evaluate our approach on three prevalent VCS, namely Vivaldi [7], Pharos [8] and Phoenix [9]. Besides being a tool for cloud users, the proposed technique could also be used by the cloud provider to increase transparency and to prove location stability of the resource.

The paper is organised as follows. First, we show in Section II related work on policy verification in cloud computing and geolocating techniques on the Internet, before Section III describes the principles of virtual network coordinates. Section IV introduces the framework for geolocating virtual resources. We show practical measurements and evaluation of

the framework in Section V and show further enhancements for the framework. Finally, we summarise and conclude the work in Section VI.

II. RELATED WORK

A. Security policy assessment in cloud computing

In cloud computing it is important in which jurisdiction the data is stored or processed because of several national regulations, e.g., the European data protection law [10]. On the other hand there are regulations that enable governmental institutions in some countries to access data stored in a cloud, e.g., the USA Patriot Act [4]. In this case the user may not want to place the data in this jurisdiction. The importance of the location of data in a cloud is also stated in [11] and [12]. In [12] the authors propose to assess the location of a cloud by a trusted third party. However, this gives only a situational view on the cloud infrastructure and the exact manual verification of the placement of user data within the cloud is difficult.

There exist some work on the enforcement of security policies and assessment in cloud computing. Especially, the cloud audit group [13] works on the automation of audit, assertion, assessment, and assurance in cloud computing. Basescu et al. [14] propose a security management framework for cloud computing for defining and enforcing security policies. However, their approach does not include the assessment of the correct implementation of security policies.

Mechanisms for technically enforcing security policies are available as well, e.g., by using encryption techniques. Iskander et al. showed in [15] a mechanism for the enforcement of authentication policies. Policy enforcement by selective resource sharing based on selective encryption is proposed by Vimercati et al. [16]. Gentry [17] works on homomorphic encryption that allows data processing on encrypted data. This approach is practically not relevant because of computational complexity and constraints on the operation. The same holds for secure multiparty computation [18], [19].

B. Geolocating Internet nodes

Ever since the Internet exist, people have been curious on finding out the geographic location behind a communication partner, which is identified by an IP address. However, the mapping of an IP address to a geographic location with a certain accuracy is not trivial. Meanwhile, several approaches have been developed, which can be basically divided into two groups: measurement based geolocation and semantic based geolocation. While the latter uses sources as *whois* or DNS queries (e.g. RFC 1876) to associate an IP address to a location, measurement based approaches use *ping* or *traceroute*. Based on these techniques, several dedicated databases are available, which offer geolocation services publicly. These databases are usually quite accurate on country-level, however as they are often proprietary and manually updated, the issue of consistency and accuracy remains [20]. Especially with an increased penetration of IPv6 addresses and consequently the dramatic increase of IP addresses, these databases may be difficult to maintain.

Initial methods of automated geolocating using latency delays have been proposed by Padmanabhan et al. [21]. Methods as in [22] improve this situation by the use of a combination of measurement based methods and semantic based approaches. However, these systems are still prone to error of up than 1000km [20]. In the same paper, Youn et al. propose a more precise statistical geolocation scheme based on kernel density estimation.

However, to our knowledge, none of these systems has been tested in a more complex network environment with additional relay nodes forwarding traffic to the actual recipient, which is assumed here. Thus, in the following, we propose an alternative approach of geolocation by the use of virtual network coordinate systems based on RTT measurements in order to identify the position of a virtual cloud resource.

III. BACKGROUND

Developed to predict latencies between nodes in the Internet without large-scale measurements, virtual network coordinate systems (VCS) also provide the ability to identify the location of nodes. Based on latency measurements they assign a physical network node to a position in an n -dimensional vector space and in that way describe its location in relation to others. Generally, the aim of such systems is to increase topology awareness and as such to optimise network traffic behaviour by predicting latency.

Based on dedicated hosts, which serve as landmarks with known locations, Global Network Positioning (GNP) [23] maps nodes in a geometric space in such a way that their distance in the VCS represents the latency between them in the physical network: $latency(A, B) \approx distance(A, B)$. It is noteworthy that this approach works for RTT measurements as well. For newly joining nodes, the positions are computed in relation to the landmarks. Obviously, this approach has the main drawback of a limited number of landmark nodes and thus missing scalability. Hence, decentralised approaches have been developed as such as Vivaldi [7] and Pharos [8]. The computation of the Vivaldi coordinate system is done by estimation of propagation times between the nodes based on the position of n randomly selected neighbours.

The precision of the VCS is usually expressed by the *prediction error*, which defines the difference between the calculated and real distance. In [8], Chen et al. showed that the prediction error on short links in Vivaldi is much higher than for long links. They therefore introduced Pharos, which uses a hierarchical structure to minimise this error. A base overlay predicts long links and attached local cluster overlays are used to predict short links. Both overlays use the Vivaldi algorithm though.

However, Vivaldi and Pharos are prone to a common behaviour in the Internet, called triangle inequality violation. The triangle inequality states that the distance between nodes A and C is less than or equal to the sum of the distances between nodes A and B and nodes B and C (see Formula 1).

$$d(A, C) \leq d(A, B) + d(B, C) \quad (1)$$

In order to avoid violations of the triangle inequality, Phoenix [9] uses a dot product based network system. Thereby, nodes are first mapped into a distance matrix, in which every entry represents the latency between nodes A and B. In a next step this matrix is factorised into two smaller matrices containing a corresponding vector for each node. The product of these two nodes' vectors then defines the distance between the nodes.

Currently, VCS's are already practically used by Peer-to-Peer applications (e.g. Vuze¹) trying to improve their performance by the use of location information.

IV. FRAMEWORK DESCRIPTION

The following section describes the settings and the architecture of our framework for detecting the location of virtual resources within a cloud.

A. Setting

The framework consists of two independent methods to verify the location of virtual cloud resources based on VCS. The first variant assumes that the cloud provider connects the cloud resource v directly to the Internet and the users wants to verify the location of his data in regard to the given policy.

In a second assumption, the cloud provider uses a relay node (i.e. proxy) in order to redirect traffic to the virtual resource in the cloud. This course of action avoids unnecessary changes for the users but also allows a movement of data to different locations without the necessity of an official notification. Fig. 1 describes the two latter sub-scenarios. Fig. 1 a) depicts the case, in which the virtual resource is located close to the relay node, within the same location, while in Fig. 1 b), the cloud provider moved the virtual resource to another location. However, both cases can be seen as the initial configuration as long as the location of v is within the boundaries of the policy-defined jurisdiction.

B. Architecture

The localisation process itself requires a set of n collaborative landmarks $K = \{k_1, k_2, \dots, k_n\}$ at distributed locations $L = \{l_1, l_2, \dots, l_m\}$. The landmarks act as reference points necessary to create meaningful coordinate systems. In the context of this work, a location is equivalent to a country, which usually defines the boundary of a jurisdiction.

Within the set of landmarks, every node is able to measure the RTT to all other nodes. For the first case, this does not necessarily imply any additional efforts for a user u , as he can reuse publicly available resources as the *King Data Set* [24] or the *Caida DNS root/gTLD RTT dataset* [25] for instance. The only requirement is that the physical positions of the nodes are known. In the case of a operating relay node, RTT measurements are still required. However due to the nature of VCS, a full set of measurements is not necessary. Vivaldi for instance computes the coordinates based on n neighbours, with $n = 64$ by default. It is noteworthy that both, latency

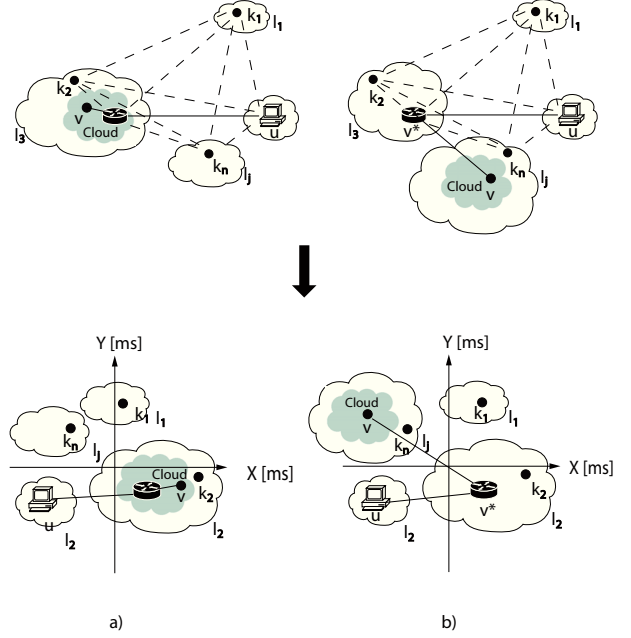


Fig. 1. Schematic description of the framework. a) The virtual resource is located in the expected cloud location. b) The cloud provider moved the virtual resource to another location, hiding the new location behind a proxy.

or RTT can be used, measurements should only be performed consistently.

Having a dataset ds available, the user u can integrate his position into this dataset as well as the position of the virtual resource v in the cloud. This is done by measuring the RTT from the two nodes between each other and to the available landmarks.

Based on this data matrix RTT_{matrix} , the user then computes the network coordinate system (Formula 2) using a prevalent algorithm like the one described in Section III.

$$VCS = f(RTT_{matrix}|u, v, ds \in RTT_{matrix}) \quad (2)$$

Presuming that the euclidean distance between nodes in a VCS maps to the *real* network distance, v being mapped to its real physical location allows its classification to a particular location l_i (i.e. in the same country). This requires a reference landmark node k_j to be placed in l_i . The prediction of v 's location is then done by the use of a supervised classification algorithm c (see Formula 3). As classification method, e.g. Instance-based learning (IB1) [26] or supported vector machine algorithm (SVM) [27] can be considered. These methods analyse the given data and predict the class for an input based on classification patterns.

¹<http://azureus.sourceforge.net/>

$$l_i = c(VCS, v) \quad (3)$$

If the real network location l_i has been determined correctly, the classification is considered to be successful.

Periodic RTT measurements will then safeguard the current position of the virtual resource. In case the cloud operator moves v to another location, the RTT is changing and finally reveals both the change of location and the new location. Thus, the user of such a service can ensure that the cloud resources are still in the policy-defined jurisdiction.

V. EVALUATION

As described in Section III, several VCS have been developed, hence three prevalent systems have been selected in order to evaluate their accuracy in geographic localisation: Vivaldi, Pharos and Phoenix.

As basis of evaluation, we used the PlanetLab network to set up a network infrastructure with known node locations and measured RTTs between 120 nodes, distributed in 28 countries and 75 Internet routable networks (LANs). Based on this selection, coordinate systems have been computed for all three systems. Concerning the number of landmarks, we simulated the location identification for 1,2,5,10,20,40,60,80 and 89 landmarks. Due to missing measurements, we had 89 nodes available to test the two approaches (direct measurements without a relay for reference and measurements through a single relay node).

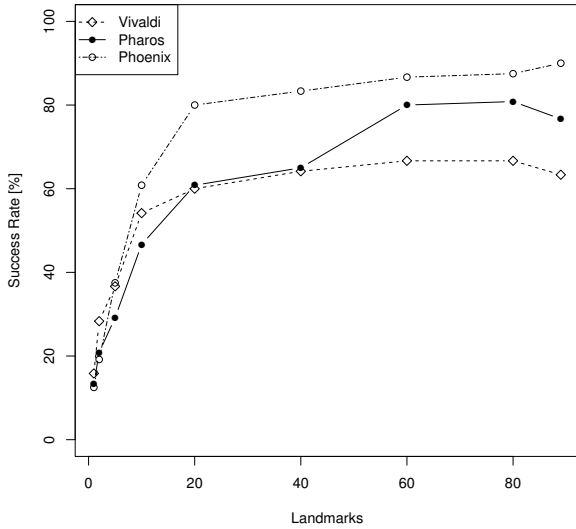


Fig. 2. Performance comparison of VCS on country level

Evaluation of the results showed that for Vivaldi and Pharos, the best results have been achieved using SVM, while the classification algorithm of IB1 performed best for Phoenix. For all systems, the classification has been done setting every node as a possible user and verifying the determined location

of v . The nodes for v and L have been selected randomly from the set of remaining nodes. Fig. 2 shows the performance of the three VCS in regard to the number of landmarks.

Without relay nodes, Phoenix outperforms the other VCS by correctly identifying the country of location of v in 90% when using the maximum number of available landmarks. The reason may be the insensitivity of Phoenix against the triangle inequality.

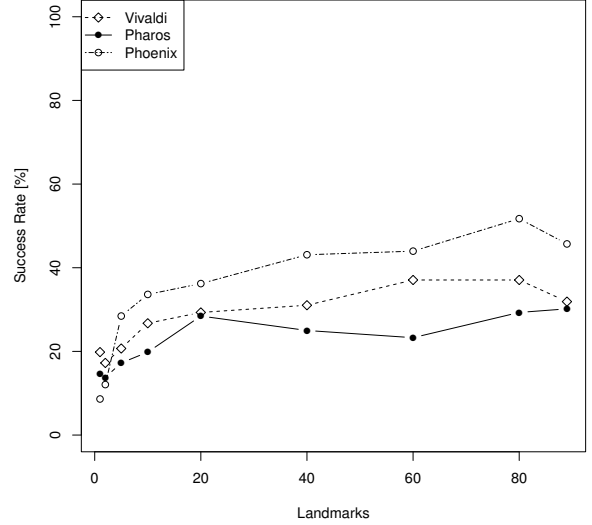


Fig. 3. Performance comparison of VCS on country level, the cloud operator using a proxy server to hide the location of v

In the second evaluation, we assume that a relay node is present all the time and used to relay traffic to another node, either located within the same geographic location or in another one. Thus all traffic to the PlanetLab nodes has been sent through a relay node, located in Amsterdam/Netherlands and again the RTT has been measured as basis for the VCS. A general increase of the RTT can be identified with a mean of 279ms and a standard deviation of 192ms.

Again, Phoenix outperforms the other VCS and achieves a precision up to 52% when using 80 landmark nodes. Even though the classification on country-level shows a significant decrease of performance, it still can give an indication of the location. A 2D plot of the Phoenix coordinate system for instance (Fig. 4), shows that most European nodes are clustered closely together, which makes geolocating on country level difficult. However, picking one of these nodes, the surrounding nodes still give a strong indication that this node is located within Europe and not moved to another continent for instance.

Another observation indicates that the performance depends strongly on the location of all involved nodes. These aspects will be considered in future work.

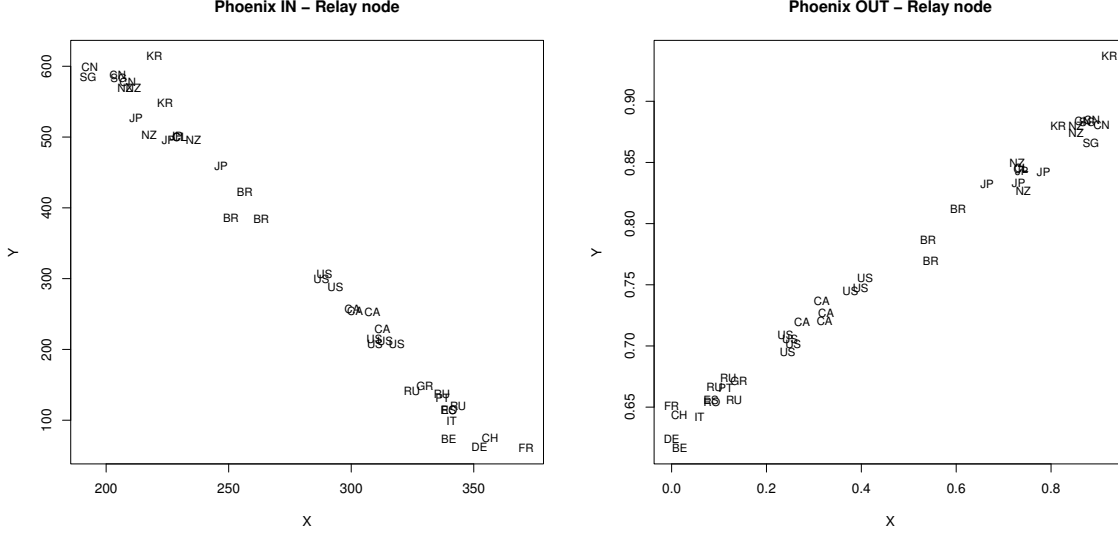


Fig. 4. Node distribution mapping of 40 nodes in the Phoenix coordinate system.

VI. CONCLUSION AND FUTURE WORK

In this paper, we showed how network coordinate systems can be used to verify the location of virtual cloud resources and to detect geographical node movements, by this being able to identify possible policy violations. Therefore, we evaluated three virtual network coordinate systems in regard to their performance of associating virtual cloud resources to geographic locations. The proposed solution highlights that VCS-based geolocation is possible even through location-concealing relay nodes without the requirement of large-scale measurements. Within the three selected coordination systems, Phoenix performs best, providing an accuracy of 52%. In addition, the VCS allows the identification of node movements, which we will quantify through future evaluation.

In the course of the evaluation, we also identified a relevant aspect with the potential to improve the achieved accuracy, namely the integration of already existing location knowledge of nodes in the neighbourhood. For instance, nodes within Europe are difficult to classify per country, but they can provide additional location information. With the help of a modified classification method and the inclusion of these nodes, we expect further significant improvement of our approach.

Another aspect of improvement is founded on the availability of publicly available datasets, which may get outdated over time. Thus, an own data set could be constructed by the help of collaborative users with the same interest in policy compliance. Users, willing to provide the geographic location of a node and willing to perform basic RTT measurements may be in a position to create a fully distributed up-to-date knowledge base.

Based on such a knowledge base, a further step will be the creation of fingerprints of the cloud datacenters. With the help of these fingerprints, users of cloud services will be in

a position to easily identify the location of their virtual cloud resource easily. A fingerprint itself could be created out of a distance profile, extracted from the systems' coordinates. This aspect will be part of future work.

During this work, we concentrated on presenting a verification method for cloud users, however, the approach may also be utilised by cloud providers towards more transparency in their processes and/or interfaces.

Finally, the VCS face certain vulnerabilities, which we did not consider in the frame of this work. These, as such as Byzantine attacks or coordinate inflation, deflation and oscillation attacks (e.g. [28], [29], [30]), are subject of distinct research.

Acknowledgements: We thank Emmanuel Nataf from INRIA Nancy - Grand Est for quickly and simply providing us access to the PlanetLab network (<http://www.planet-lab.org>), which allowed us to conduct the practical evaluation.

REFERENCES

- [1] "Amazon Virtual Private Cloud," July 2011. [Online]. Available: <http://aws.amazon.com/ec2/>
- [2] "Google App Engine," July 2011. [Online]. Available: <http://code.google.com/appengine/>
- [3] "Google Docs," July 2011. [Online]. Available: <http://docs.google.com>
- [4] D. Fraser, "The canadian response to the USA Patriot Act," *Security Privacy. IEEE*, vol. 5, no. 5, pp. 66–68, sept.-oct. 2007.
- [5] "IP Address Location," August 2011. [Online]. Available: <http://www.ipaddresslocation.org/>
- [6] "IP Address Geolocation to Identify Website Visitor's Geographic Location," August 2011. [Online]. Available: <http://www.ip2location.com/>
- [7] F. Dabek, R. Cox, M. F. Kaashoek, and R. Morris, "Vivaldi: a decentralized network coordinate system," in *SIGCOMM*, 2004, pp. 15–26.
- [8] Y. Chen, Y. Xiong, X. Shi, B. Deng, and X. Li, "Pharos: A decentralized and hierarchical network coordinate system for internet distance prediction," in *GLOBECOM*, 2007, pp. 421–426.

- [9] Y. Chen, X. Wang, X. Song, E. K. Lua, C. Shi, X. Zhao, B. Deng, and X. Li, "Phoenix: Towards an accurate, practical and decentralized network coordinate system," in *Networking*, 2009, pp. 313–325.
- [10] "Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (directive on privacy and electronic communications)," in *Official Journal of the European Union*, no. L201, 2002, pp. 0037–0047.
- [11] ENISA, "Cloud computing security risk assessment," European Network and Information Security Agency (ENISA), Tech. Rep., 2009.
- [12] J. Heiser and M. Nicolett, "Assessing the security risks of cloud computing," Gartner, Tech. Rep., 2008.
- [13] "CloudAudit: A6 - The Automated Audit, Assertion, Assessment, and Assurance API," July 2011. [Online]. Available: <http://cloudataudit.org>
- [14] C. Basescu, A. Carpen-Amarié, C. Leordeanu, A. Costan, and G. Antoniu, "Managing data access on clouds: A generic framework for enforcing security policies," in *AINA*. IEEE Computer Society, 2011, pp. 459–466.
- [15] M. K. Iskander, D. W. Wilkinson, A. J. Lee, and P. K. Chrysanthos, "Enforcing policy and data consistency of cloud transactions," in *Proceedings of the Second International Workshop on Security and Privacy in Cloud Computing*, ser. ICDCS-SPCC 2011. Washington, DC, USA: IEEE Computer Society, 2011.
- [16] S. D. C. d. Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, G. Pelosi, and P. Samarati, "Encryption-based policy enforcement for cloud storage," in *Proceedings of the 2010 IEEE 30th International Conference on Distributed Computing Systems Workshops*, ser. ICDCSW '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 42–51. [Online]. Available: <http://dx.doi.org/10.1109/ICDCSW.2010.35>
- [17] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *STOC '09: Proceedings of the 41st annual ACM symposium on Theory of computing*. New York, NY, USA: ACM, 2009, pp. 169–178.
- [18] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, pp. 612–613, November 1979.
- [19] A. C. Yao, "Protocols for secure computations," in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, ser. SFCS '82. Washington, DC, USA: IEEE Computer Society, 1982, pp. 160–164.
- [20] I. Youn, B. L. Mark, and D. Richards, "Statistical geolocation of internet hosts," *Computer Communications and Networks, International Conference on*, vol. 0, pp. 1–6, 2009.
- [21] V. N. Padmanabhan and L. Subramanian, "An investigation of geographic mapping techniques for internet hosts," in *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '01. New York, NY, USA: ACM, 2001, pp. 173–185. [Online]. Available: <http://doi.acm.org/10.1145/383059.383073>
- [22] E. Katz-Bassett, J. P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and Y. Chawathe, "Towards ip geolocation using delay and topology measurements," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, ser. IMC '06. New York, NY, USA: ACM, 2006, pp. 71–84. [Online]. Available: <http://doi.acm.org/10.1145/1177080.1177090>
- [23] T. S. E. Ng and H. Zhang, "Towards global network positioning," in *Proceedings of the First ACM SIGCOMM Workshop on Internet Measurement*, 2001, pp. 25–29.
- [24] "King : A tool to estimate latency between any two Internet hosts, from any other Internet host." 2002. [Online]. Available: <http://www.mpi-sws.org/~gummadi/king/>
- [25] "The CAIDA DNS root/gTLD RTT Dataset," August 2011. [Online]. Available: http://www.caida.org/data/passive/dns_root_gtld_rtt_dataset.xml
- [26] D. W. Aha and D. Kibler, "Instance-based learning algorithms," in *Machine Learning*, 1991, pp. 37–66.
- [27] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [28] D. Zage and C. Nita-Rotaru, "On the accuracy of decentralized network coordinate systems in adversarial networks," in *In The 14th ACM Conference on Computer and Communications Security (CCS)*, 2007. [Online]. Available: <http://www.cs.purdue.edu/homes/zagedj/docs/ccs050-zage.pdf>
- [29] D. J. Zage and C. Nita-Rotaru, "On the accuracy of decentralized virtual coordinate systems in adversarial networks," in *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2007, pp. 214–224.
- [30] S. Becker, J. Seibert, D. Zage, C. Nita-Rotaru, and R. State, "Applying game theory to analyze attacks and defenses in virtual coordinate systems," in *International Conference on Dependable Systems and Networks (DSN)*, 2011.