Secure communication using dynamic VPN provisioning in an Inter-Cloud environment

Ali Sajjad, Andrea Zisman, Muttukrishnan Rajarajan
City University London
EC1V 0HB London, UK
Email: {Ali.Sajjad.1, A.Zisman, R.Muttukrishnan}@city.ac.uk

Srijith K. Nair, Theo Dimitrakos
British Telecom, Innovate and Design
BT Adastral Park, Ipswich, UK
Email: {srijith.nair, theo.dimitrakos}@bt.com

***Abstract*—Most of the current cloud computing platforms offer Infrastructure as a Service (IaaS) model, which aims to provision basic virtualised computing resources as on-demand and dynamic services. Nevertheless, a single cloud does not have limitless resources to offer to its users, hence the notion of an Inter-Cloud enviroment where a cloud can use the infrastructure resources of other clouds. However, there is no common framework in existence that allows the srevice owners to seamlessly provision even some basic services across multiple cloud service providers, albeit not due to any inherent incompatibility or proprietary nature of the foundation technologies on which these cloud platforms are built. In this paper we present a novel solution which aims to cover a gap in a subsection of this problem domain. Our solution offer a security architecture that enables service owners to provision a dynamic and service-oriented secure virtual private network on top of multiple cloud IaaS providers. It does this by leveraging the scalability, robustness and flexibility of peer-to-peer overlay techniques to eliminate the manual configuration, key management and peer churn problems encountered in setting up the secure communication channels dynamically, between different components of a typical service that is deployed on multiple clouds. We present the implementation details of our solution as well as experimental results carried out on two commercial clouds.**

I. Introduction

Most of the currently available Cloud Computing solutions are mainly focused on providing functionalities and services at the infrastructure level, e.g., improved performance for virtualization of compute, storage and network resources, as well as necessary fundamental functionality such as virtual machine (VM) migrations and server consolidation etc. In the cases when higher-level and more abstract concerns need to be addressed, existing Infrastructure as a Service (IaaS) solutions tend to focus on functional aspects only. Furthermore, if a cloud's computational and storage infrastructure resources are overloaded due to increased workloads, its service towards it clients will degrade. The idea of an Inter-Cloud [1] has been gaining much traction to address such a situation, where a cloud can borrow the required infrastructure resources of other clouds. However, in order to progress from a basic cloud service infrastructure to a more adaptable cloud service ecosystem, there is a great need for tools and services that support and provide higher-level concerns and non-functional aspects in a comprehensive manner.

The OPTIMIS project [2] is an ongoing effort in this regard which strives to provide a holistic approach to cloud service provisioning by offering a single abstraction for multiple coexisting cloud architectures. Of the various high-level concerns being addressed by the OPTIMIS project, a major concern of high importance is the provisioning of a secure communication framework to the services utilizing the resources of different cloud IaaS providers. The usage pattern of these services is usually quite flexible i.e. on one hand they might be directly accessed by end-users or on the other hand they might be orchestrated by other Service Providers (SP) for their customers.

There are three fundamental steps in the life cycle of a service in the cloud computing ecosystem; the construction of the service, the deployment of the service to one or more IaaS clouds and lastly the operational management of the service. In the resulting scenarios, the presence of the multiple IaaS providers in the cloud ecosystem is the key issue that needs to be addressed by any inter-cloud security solution. A major goal of service owners is to select IaaS providers in an efficient way in order to host the different components of their services on appropraite clouds. In this respect, third-party cloud brokers [3] can play a major role in simplifying the use, performance and delivery of the cloud services. These brokers can also offer an inter-mediation layer spanning across multiple cloud providers to deliver a host of optimisation and value-added services which take advantage of the myriad individual cloud services e.g., aggregation of different services or arbitration for a best-match service from multiple similar services. For the numerous interaction possibilities among these parties, whatever the usage scenarios maybe, the security of data and the communication between the consumers of the service and its multiple providers is of paramount importance.

In the light of the above discussion, it is clear that an inter-cloud security solution is highly desirable that would provide a framework enabling seamless and secure communication between the actors of a cloud ecosystem over multiple cloud platforms. Such a solution, however, has to overcome a number of challenges because of architectural limitations. This is because most of the current cloud service platforms, and the multi-tenants environments they offer, make it difficult to give the consumers of their services flexible and scalable control over the core security aspects of their services like encryption, communication isolation and key management. Secure communication is also challenged by lack of dynamic network configurability in most cloud providers, caused by the inherent limitations of the fixed network architectures offered by these providers.

In this work we address the secure, flexible and scalable communication concerns that in our view must be overcome in order to provide holistic provisioning of services to consumers from multiple cloud service providers. We present the architecture and design of an inter-cloud secure communication framework that offers the features of dynamic and scalable virtual network formation, efficient and scalable key management and minimal manual configuration all on top of secure and private communication between the components of the service across multiple cloud platforms. Our architecture provides a single virtual network to the service using resources from multiple cloud providers and offers the capability to efficiently and transparently run services on top of this network while catering for the dynamic growth and shrinkage of the components of the service.

The rest of the paper is organised as follows: In Section *II* we outline the key motivations for our approach. In Section *III* we present the background and related works that address peer-to-peer overlays, virtual network connectivity and key management issue related to this domain. We elaborate on the detailed Inter-Cloud Virtual Private Network architecture in Section *IV*. In Section *V* we present our experimental setup and the analysis of the performance results of our solution. We conclude in Section *VI* with the future directions of our work.

II. Motivation

The design and architecture of our inter-cloud secure communication framework is inspired by a collection of techniques like Virtual Private Networks [4] (VPN) and Peer-to-Peer (P2P) Overlays [5]. Network virtualization techniques like VPNs and P2P Overlays have been shown to provide their users legacy communication functionalities of their native network environments, despite the topology, configuration and management architecture of the actual underlying physical network. This fits perfectly with our goal of providing a secure virtual private network as a service to the consumers operating on top of multiple cloud providers. All complications and complexities of managing a physical network can be handled by the overlay network, enabling the services deployed on multiple clouds to benefit from a customised communication network typically only available in physical local-area environments.

Traditionally, most of the private network solutions for similar problem spaces require the direct and continuous control of a centralised administration entity over every aspect of the overlay network, consisting of all the participants that constitute and facilitate the operation of the service being deployed and run on the multiple cloud providers. Such a central controller provides services to authenticate, secure and police the interactions amongst peers. These centralized solutions make it almost necessary to provide complex support and management functionalities to meet the user demands of smooth and continuous operation. Furthermore, to robustly handle the loads generated by a large number of users, significant infrastructure resources and services like mirroring or redundant instances and load-balancers must be set aside, incuring additional costs for the service owner. Peer-to-peer overlays, on the other hand, are designed to offer improved scalability, flexibility and availability in a distributed fashion without extensive reliance on centralized servers or resources. For these reasons, such overlay networks have been used very successfully to provide specialized application layer services like voice over IP (VoIP) e.g., Skype [6] and file sharing e.g., Bittorrent [7]. Structured P2P overlay networks based on distributed hash tables (DHT) support the scalable storage and retrieval of *key*, *value* pairs on the overlay network which is very helpful when we need to store and retrieve meta-data related to the virtual private network management. Existing P2P algorithms like Chord [8], Pastry [9] and Tapestry [10] have been widely used to provide scalable and fast information storage and retrieval services for a vast variety of applications. We have leveraged the Kademlia algorithm [11] to cater for our storage and retrieval requirements to build up a virtual private network. This DHT-based algorithm locates values using the peer ID and guarantees that on average, any data object can be located in $O\ (log\ N)$ peer hops, N being the number of peers in the overlay.

Therefore, by provisioning a VPN among the nodes of a P2P overlay network, we can enable feature of using secure communication between the components of a service deployed on multiple clouds. Furthermore, we promote an approach where a distributed and scalable key management framework is utilized to provide the cryptographic primitives used to establish secure tunnels among the nodes of the P2P overlay networks. The synergy of these three technologies produces a scalable, secure and robust inter-cloud communication solution which is able to handle a large number of communicating peers with considerably less management complexity.

In this paper, we present the design and architecture of such an Inter-Cloud Virtual Private Network (ICVPN) solution, which provides secure communication facilities to service owners that want to deploy their service components over the infrastructure of multiple cloud IaaS providers. At its core, it provides the ability to automatically establish peer-to-peer overlay networks among the virtual machines (VMs) constituting the cloud service. Using the same P2P techniques, we also offer a distributed key management service which facilitates the binding of cryptographic constructs like keys, certificates and tokens to the VMs constituting the service. The configuration and maintenance of the VPN connections using the P2P overlay is autonomous and transparent to the service, as a major goal of our work is to free the service owner from the complicated configurations typically required to set up the key management and virtual networking infrastructures in similar problem spaces.

III. Related Work

The central thrust of our architecture is the provisioning of a secure virtual private network over multi-cloud infrastructure. VPNs have been a mainstay for providing secure remote access over wide-area networks to resources in private organizational networks for a long time. Well-known tools and softwares like OpenVPN [12] are used to create secure point-to-point or site-to-site connections for authenticated remote access. However, the main problem in client/server based approaches is that they require centralized servers to manage the life cycle of all the secure connections for the participating clients, hence suffering from a single point-of-failure. Another issue is the quite complex and error prone configuration problems especially if you want to construct and manage a large-scale network not having a relatively simple topology, as it would require customised configuration on every client and even more elaborate management and routing configuration on the server-side. Another major drawback is the complexity of key distribution among all the participating clients in a VPN, as the software itself does not provide any key distribution service and all keys have to be manually transferred to individual hosts. In case of PKI model, an additional requirement of a trusted Certificate Authority exists that has to issue individual certificates to all the servers and clients constituting a VPN, which incurs an additional communication overhead when forming a virtual private network.

There have been some other VPN solutions for large-scale networks aimed at grid and cluster computing environments, such as VIOLIN [13] and VNET [14], that do not follow a strict client/server model based approach. VNET is a layer 2 virtual networking tool that relies on a VNET server running on a Virtual Machine Monitor (VMM) hosting a virtual machine in a remote network which establishes an encrypted tunnel connection to a VNET server running on a machine (called Proxy) inside the users home network. All of the remote virtual machines communication goes through this tunnel and the goal of the Proxy is to emulate the remote virtual machine as a local host on the users home network, in effect presenting it as a member of the same LAN. The motivation of this approach is to tackle the users lack of administrative control at remote grid sites to manipulate network resources like routing and resource reservations etc. but it suffers from the previously discussing problem of complex and manual configuration though going for the simplicity of a private LAN. Also the scalability will be a big issue for the Proxy as the number of remote virtual machines grows as each will require a secure tunnel connection and corresponding virtual network interface mapped to the Proxys network interface by the VNET server software.

VIOLIN is a small-scale virtual network with virtual routers, switches and end hosts implemented in software and hosted by User-Mode Linux (UML) enabled machines as virtual appliances. It allows for the dynamic establishment of a private layer 3 virtual network among virtual machines, however, it doesnt offer dynamic or automatic network deployment or route management to setup the virtual network. Virtual links are established between the virtual appliances using encrypted UDP tunnels that have to be manually setup and are not self-configuring, making it cumbersome to establish inter-host connections in flexible and dynamic fashion.

P2P VPN solutions like Hamachi [15] and N2N [16] have come up as peer-to-peer alternatives to centralized and client/server model based VPNs. Hamachi is a shareware application that is capable of establishing direct links between computers that are behind NAT firewalls. A backend cluster of servers are used to enable NAT traversal and establish direct peer-to-peer connections among its clients. Each client establishes and maintains a control connection to the server cluster. It is mainly used for internet gaming and remote administration but suffers from scalability issues as each peer has to maintain the connection with the server as well as any other peers it wants to communicate with, ending up with the overhead of a mesh-topology. It therefore offers limited number of peers (16 per virtual network) and limited number of concurrent clients (50 per virtual network). The keys used for connection encryption and authentication are also controlled by the vendors servers and individual users do not initially control who has access to their network. N2N is a layer 2 VPN solution which doesnt require a centralized backend cluster of servers like Hamachi but it uses a peer-to-peer overlay network similar to Skype, where a number of dedicated super-nodes are used as relay agents for edge nodes that cannot communicate directly with each other due to firewall or NAT restrictions. The edge nodes connect to a super-node at start-up and pre-shared TwoFish [17] keys are used for link encryption. As it operates on layer 2, the users of the overlay have to configure their IP addresses etc. It also assumes node membership as relatively static with edge nodes rarely leaving or joining the network over their life cycle.

More recently, some commercial cloud computing services have been made available by different vendors that provide a virtual private network inside their public cloud offering and offering the customers some limited degree of control over this network, which is called a Virtual Private Cloud (VPC). Prime examples in this domain are Amazon Virtual Private Cloud [18], Google Secure Data Connector [19] and CohsiveFT VPN-Cubed [20]. These are aimed at enterprise customers to allow them to access their resource deployed on the vendors cloud over an IPSec [21] based virtual private network. Although these products allow the possibility of leveraging the cloud providers APIs to flexibly grow and shrink their networks, the management and configuration is as complex as a traditional network as components of the VPC such as internet gateways, VPN servers, NAT instances and subnets have to be managed by the customers themselves. Furthermore, the customers are required to setup an IPSec device on their premises that connects to an IPSec gateway in the VPC running as a virtual appliance which integrates the enterprises network with the VPC subnet in the cloud. Most importantly, with the exception of [20], these solutions are locked to single cloud vendor and [20] provides use of a selective set of cloud providers by placing its virtual appliances as VPN gateways in these cloud infrastructures and allowing the customers to join these gateways in a mesh topology manually.

IV. Inter-Cloud VPN Architecture

In this section we present the inter-Cloud VPN architecture (ICVPN) that we are proposing. The architecture consists of two main components, namely *(a)* the peer-to-peer overlay and *(b)* the secure virtual private connections, as described below.

A. Peer-to-Peer Overlay

The core technique employed by the ICVPN concept is the use of two tiers of P2P overlays. A universal P2P overlay is used to provide a scalable and secure service infrastructure to initiate and bind multiple VPN overlays to different cloud services. The universal overlay itself can be initiated either by the service owner, the cloud broker or the cloud service providers. It helps with the bootstrapping activity of VPN peers. It also provides other functionalities such as service advertisement, service discovery mechanisms, and service code provisioning, with minimal requirement for manual configuration and administration.. This approach acts as an aggregation service for the eventual peered overlay resources (which in this case are virtual machines) span across multiple cloud domains to help form a virtual private network. The peers of the universal overlay act as super peers for the nodes of the underlying overlays and let new nodes enroll, authenticate, bootstrap and join a particular VPN overlay based on the cloud service requiring a VPN service.

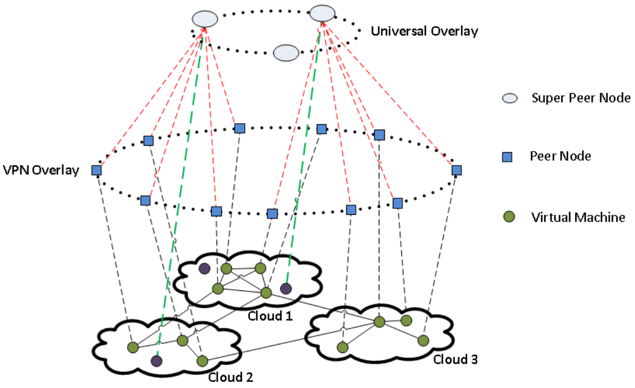

Fig. 1. Two-tiered architecture for the Inter-Cloud VPN solution

As depicted in Fig. 1, the service owner/provider or the cloud broker could itself be a peer in the universal overlay and a subset of the universal overlay peers can act as super-peers for the peer nodes of the VPN overlay for a particular cloud service. The universal overlay peers can join and leave the system dynamically and additional VMs from the cloud providers can be provisioned to act as the universal overlay peers as well. As both the universal and the VPN overlay nodes are basically VMs provisioned from different cloud providers, they can be demoted or promoted from these overlays respectively based on parameters like performance and availability.

To join the universal overlay, each peer needs to acquire a unique identification number (peerID). In most structured P2P systems, this is done by the peer itself by choosing a random number from a large identity space, however, this approach is vulnerable to Sybil attacks [22]. Due to the security constraints of our solution, we require some trusted authorities to allocate peerIDs to the participating peers. We solve this identity management problem by using Trusted Third Party (TTP) model to authenticate peers and allocate them their identities. We make use of the traditional PKI approach and designate the super-peers of the Universal Overlay as Certificate Authorities (CA) for the underlying VPN overlays peers. The CA assigns peerIDs to the peers and signs a certificate that binds the ID of the cloud service utilizing the VPN (serviceID) and peerID within the public certificate of the peer for a limited time duration. The peer then can use this signed certificate to authenticate itself with other peers in the overlay.

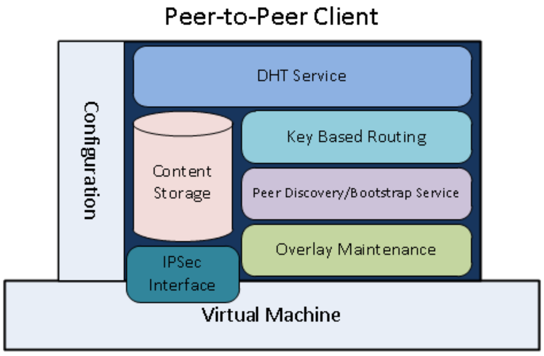

Fig. 2. Architecture of a P2P client in the VPN overlay

In a typical usage scenario, the service owner is responsible for provisioning virtual machines from cloud service providers to deploy and run their services. These virtual machines are considered as the peers of the VPN overlays and the complete life-cycle of the peers is handled by a P2P client embedded in the appliance image used to instantiate a virtual machine on a cloud platform. However, a further advantage of the universal overlay approach is that the peers of a VPN overlay can get, update and modify the P2P client program dynamically from the super-peers in the universal overlay. The program to be run is signed by the super-peers for validity and it can check for updated versions of itself by querying for the associated serviceID in the persistent store of the universal overlays DHT.

B. Secure Virtual Private Connections

The key feature of our ICVPN is establishing a secure communication network between the peers of the overlay formed over a collection of cloud providers infrastructure. Therefore, after successfully joining the overlay network to become part of a service, a VPN peer starts the process of creating secure tunnels to the other peers of the service it wants to communicate with, according to the functional operations of that particular service. To achieve this, we make use of IPSec [23] to authenticate and encrypt each IP packet of a communication session between the peers, thus creating end-to-end tunnels which provide protection against eavesdropping, message tempering and message forgeries. For establishing mutual authentication between peers at the beginning of the session and negotiation of cryptographic keys to be used during the session, we employ the Internet Key Exchange protocol [24], which makes use of standard cryptographic primitives like public key cryptography [25] for establishing mutual authentication and AES [26] for the actual encryption of packets in transition. The practical advantage of this approach is the reuse of existing frameworks and tools which have been thoroughly tried and tested in a myriad of different domains, are widely used and have been adopted in both academic and commercial domain. The main components of the P2P client used to construct a virtual private network topology in our model are shown in Fig. 2.

The P2P client software sets up and configures the IPSec security associations according the the service network security policy, which is advertised by the service owner through the DHT of the Universal Overlay. The peers of the underlying VPN overlay periodically check for any update in the security policy and apply and enforce any changes on the kernel of the VM through the P2P client's IPSec interface.

V. Performance Evaluation

In this section we present the results of a series of experiments we conducted to evaluate the effect of our prototype ICVPN solution upon the network performance of a service deployed on two different cloud IaaS providers. We use a 3-tier web service comprising of database, business logic and presentation components deployed on nine virtual machines hosted on the clouds of British Telecom Ltd. and Flexiant Ltd., our partners in the EU OPTIMIS project. The purpose of these experiments is to evaluate the architecture being proposed, in terms of service latency and service throughput, in a practical scenario with a service deployed over a real wide-area network, with the BT cloud geographically located in Ipswich, England and Flexiant cloud located in Livingston, Scotland. We define service latency as the inter-cloud round-trip time taken by a HTTP request, issued by a service component on one cloud, to get a response from the target service component on a different cloud. Similarly, service throughput is the inter-cloud network throughtput between service components deployed on different clouds.

A. Analysis: Service Latency

We compare the latency between the components of the service deployed on different cloud providers, as the latency between the components in the same cloud is almost negligible as they are usually hosted on the same hyper-visor. We measured the latency by using the round-trip delay of an HTTP HEAD request/response pair, as the components of the web service communicate with each other using HTTP protocol and ICMP, the de facto latency measurement protocol, is blocked in the networks of our cloud providers. We computed the average latency by running 10 experiments very hour for a period of 24 hours, firstly without using the ICVPN solution and then with it. The results are shown in Fig. 3.

Looking at the results, we can see that using our solution only has a small impact on the HTTP latency, increasing it just by about 5%. For ease of analysis, we collect the network traffic dump when running our experiments, using the tcpdump packet sniffer. We found out from the traffic dumps that the increased delay we encountered is mostly due to the additional packets tranmitted and received by the peers for the purposes of key exchange and cryptographic primitives negotitation when establishing an IPSec tunnel. After this initial handshake phase is over, the latency performance is almost same in the comparitive experiments.

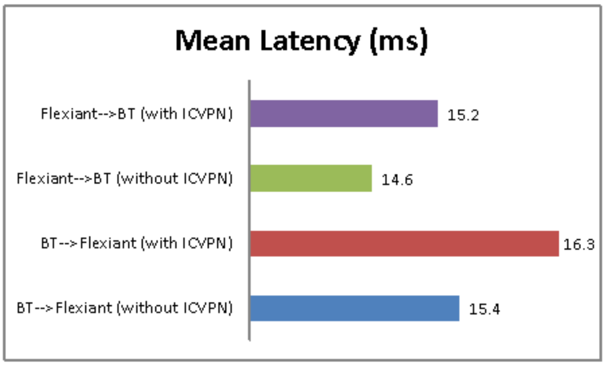

Fig. 3. Mean latency of 240 round-trip time experiments in both directions between BT and Flexiant clouds

B. Analysis: Service Throughput

We measure the throughput between components of the service deployed on different cloud providers by using Iperf [27], a commonly used network testing tool. We measured the throughput in both directions by transferring 30 MB data, a size chosen empirically to saturate the WAN links between the components and get the throughput results representing realistic conditions. We computed the average throughput by running 10 experiments every hour for a period of 24 hours, firstly without using the ICVPN solution and then applying the security policy to tunnel the traffic through IPSec. The results are shown in Fig. 4.

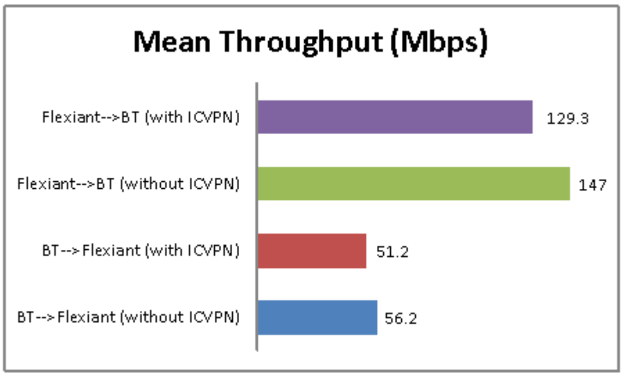

Fig. 4. Mean throughput of 240 data transmission experiments in both directions between BT and Flexiant clouds

From the throughput results, the first thing that stands out is the difference in the throughput values depending on the direction of transferring the data. Although we don't have the detailed knowledge of the underlying physical wide-area network connectivity between the two cloud service providers, such readings are not unheard of in this domain and are usually due to differences in upstream and downstream traffic conditions, different routes chosen by the IP packets or network configuration issues. Irrespective of that, by looking at the comparative results it is clear that we just incur a small overhead in the throughput, of about 10%. By analysing the traffic dumps generated from the throughout test, we can attribute this overhead to the IKE and IPSec handshakes in addition to the extra time taken by the VM kernel in encrypting and encapsulating 30 MB of data for each throughput test.

VI. Conclusion and Future Work

In this paper, we present a scalable and robust secure communication framework for services deployed in an inter-cloud environment. We employ the flexibility and scalability afforded by structure peer-to-peer overlays to join virtual machines running on different cloud IaaS providers with each other using IPSec tunnels, hence providing confidentiality, authentication and integrity for all the data exchanged between different components of the service. Our solution needs minimal manual configuration as peers are automated to discover the information needed to perform their operations from the Universal Overlay. We also provide a distributed and scalable key management solution for the consumption of the virtual machines to set-up the secure communication channels. Our solution supports the dynamic addition and removal of nodes from the VPN overlay as we use the peer-to-peer DHT not just as a command and control channel for managing the VPN peers but also for the churn management of peers in the VPN overlay. We have evaluated a prototype implementation based on experiments conducted in realistic conditions, over multiple cloud infrastructure environments and found minimal latency and throughput overhead of creating and maintaining the ICVPN connections among the participating VMs of a service.

In the future, we aim to provide a federated identity management solution utilising the ICVPN architecture in conjunction with Hierarchical Identity-Based Cryptography (HIBC) [28].

Acknowledgement

We acknowledge financial support for this work provided by the European Commission's Seventh Framework Programme ([FP7/2001-2013]) under grant agreement number 257115, OPTIMIS.

References

[1] R. Buyya, R. Ranjan, and R. N. Calheiros, "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services," in *Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2010*, 2010.

[2] A. J. Ferrer, F. Hernndez, J. Tordsson, E. Elmroth, C. Zsigri, R. Sirvent, J. Guitart, R. M. Badia, K. Djemame, and W. Ziegler, "OPTIMIS: a holistic approach to cloud service provisioning," in *First International Conference on Utility and Cloud Computing*, Dec. 2010.

[3] Gartner, "Cloud consumers need brokerages to unlock the potential of cloud services," Jul. 2009. [Online]. Available: http://www.gartner.com/it/page.jsp?id=1064712

[4] A. S. Tanenbaum and D. J. Wetherall, "Virtual private networks," in *Computer Networks*, 5th ed. Prentice Hall, Oct. 2010, p. 821.

[5] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," *SIGCOMM Comput. Commun. Rev.*, Jan. 2002.

[6] S. Baset and H. Schulzrinne, "An analysis of the skype peer-to-peer internet telephony protocol," *CoRR*, 2004.

[7] B. Cohen, "The BitTorrent protocol specification," 2001. [Online]. Available: http://www.bittorrent.org/beps/bep_0003.html

[8] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *ACM SIGCOMM*, 2001.

[9] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for Large-Scale Peer-to-Peer systems," in *Middleware 2001*, 2001.

[10] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, "Tapestry: a resilient global-scale overlay for service deployment," *Selected Areas in Communications, IEEE Journal on*, Jan. 2004.

[11] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the xor metric," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*. Springer-Verlag, 2002.

[12] J. Yonan, "OpenVPN - an open source SSL VPN solution." [Online]. Available: http://openvpn.net/

[13] X. Jiang and D. Xu, "VIOLIN: virtual internetworking on overlay INfrastructure," in *In Proc. Of The 2nd Intl. Symposium On Parallel And Distributed Processing And Applications*, 2003.

[14] A. I. Sundararaj and P. A. Dinda, "Towards virtual networks for virtual machine grid computing," in *In Proceedings of the 3rd USENIX Virtual Machine Research And Technology Symposium*, 2004.

[15] "Hamachi - a zero-configuration virtual private network." [Online]. Available: https://secure.logmein.com/products/hamachi2

[16] L. Deri and R. Andrews, "N2N: a layer two Peer-to-Peer VPN," in *Resilient Networks and Services*, 2008.

[17] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, *The Twofish encryption algorithm: a 128-bit block cipher*. New York, NY, USA: John Wiley & Sons, Inc., 1999.

[18] Amazon, "Virtual private cloud." [Online]. Available: http://aws.amazon.com/vpc

[19] Google, "Secure data connector." [Online]. Available: http://code.google.com/securedataconnecto

[20] CohesiveFT, "VPN-Cubed." [Online]. Available: http://www.cohesiveft.com/vpncubed

[21] N. Doraswamy, *IPSec : the new security standard for the Internet, intranets, and virtual private networks*, 2nd ed. Prentice Hall PTR, 2003.

[22] J. Douceur, "The sybil attack," in *Peer-to-Peer Systems*. Springer Berlin / Heidelberg, 2002.

[23] R. Atkinson, "Security architecture for the internet protocol," in *RFC 1825*, 1995.

[24] D. Harkins, "Internet key exchange (ike)," in *RFC 2409*, 1998.

[25] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, Nov. 1976.

[26] F. I. P. S. P. 197, "Announcing the advanced encryption standard (aes)," 2001.

[27] L. C. Ajay Tirumala and T. Dunigan, "Measuring end-to-end bandwidth with iperf using web100," in *Web100, Proc. of Passive and Active Measurement Workshop*, 2003.

[28] C. Gentry and A. Silverberg, "Hierarchical ID-Based cryptography," in *Advances in Cryptology ASIACRYPT 2002*, 2002.