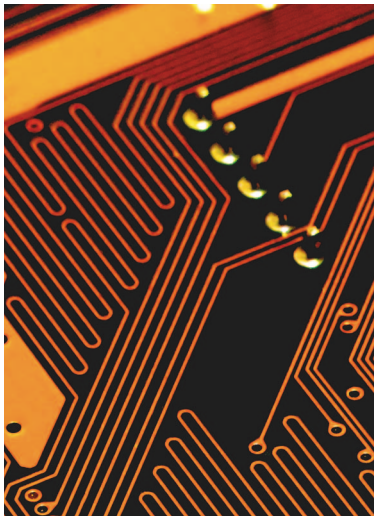
Collaboration in Multicloud Computing Environments: Framework and Security Issues



Mukesh Singhal and Santosh Chandrasekhar, *University of California, Merced*
Tingjian Ge, *University of Massachusetts Lowell*
Ravi Sandhu and Ram Krishnan, *University of Texas at San Antonio*
Gail-Joon Ahn, *Arizona State University*
Elisa Bertino, *Purdue University*

A proposed proxy-based multicloud computing framework allows dynamic, on-the-fly collaborations and resource sharing among cloud-based services, addressing trust, policy, and privacy issues without preestablished collaboration agreements or standardized interfaces.

The recent surge in cloud computing arises from its ability to provide software, infrastructure, and platform services without requiring large investments or expenses to manage and operate them. Clouds typically involve service providers, infrastructure/resource providers, and service users (or clients). They include applications delivered as services, as well as the hardware and software systems providing these services.

Cloud computing characteristics include a ubiquitous (network-based) access channel; resource pooling; multitenancy; automatic and elastic provisioning and release of computing capabilities; and metering of resource usage (typically on a pay-per-use basis).[1] Virtualization of resources such as processors, network, memory, and storage ensures scalability and high availability of computing capabilities. Clouds can dynamically provision these virtual resources to hosted applications or to clients that use them to develop their own applications or to store data. Rapid provisioning and dynamic reconfiguration of resources help cope with variable demand and ensure optimum resource utilization.

As more organizations adopt cloud computing, cloud service providers (CSPs) are developing new technologies to enhance the cloud's capabilities. Cloud mashups are a recent trend; mashups combine services from multiple clouds into a single service or application, possibly with on-premises (client-side) data and services. This service composition lets CSPs offer new functionalities to clients at lower development costs. Examples of cloud mashups and technologies to support them include the following:

- *IBM's Mashup Center*, a platform for rapid creation, sharing, and discovery of reusable application building blocks (like widgets and feeds) with tools to easily assemble them into new Web applications.
- *Appirio Cloud Storage*, a cloud-based storage service that lets Salesforce.com cloud customers store information about accounts, opportunities, and so on in the Amazon S3 cloud.
- *Force.com* for the Google App Engine, a set of libraries that enable development of Web and business applications using resources in the Salesforce.com and Google clouds.

 Published by the IEEE Computer Society

Collaboration among multiple cloud-based services, like cloud mashups, opens up opportunities for CSPs to offer more-sophisticated services that will benefit the next generation of clients.

For example, cloud-based electronic medical record (EMR) management systems like Practice Fusion, Verizon Health Information Exchange, Medscribbler, and GE Healthcare Centricity Advance are emerging. In addition, government agencies are working toward building interoperable healthcare information systems that promote electronic exchange of data across multiple organizations. These developments will influence healthcare providers to interact with multiple cloud-based EMR systems in the future.

Today, cloud mashups require preestablished agreements among providers as well as the use of custom-built, proprietary tools that combine services through low-level, tightly controlled and constraining integration techniques. This approach to building new collaborative services does not support agility, flexibility, and openness. Realizing multicloud collaboration's full potential will require implicit, transparent, universal, and on-the-fly interaction involving different services spread across multiple clouds that lack preestablished agreements and proprietary collaboration tools.

The research community is beginning to develop architectures, technologies, and standards to support collaboration among multiple cloud systems.[2-5] However, these research proposals still remain constraining due to their provider-centric approach or limited scope.

As the name suggests, provider-centric approaches require CSPs to adopt and implement the changes that facilitate collaboration—changes such as standardized interfaces, protocols, formats, and other specifications, as well as new architectural and infrastructure components. Without these provider-centric changes, current proposals do not provide facilities for client-centric, on-the-fly, and opportunistic combinations of heterogeneous cloud-based services.

While cloud standardization will promote collaboration, there are several hurdles to its adoption.[6,7] From a market perspective, it is unlikely that multiple CSPs will agree on an easy and standardized way to access services, as this would give clients total freedom in changing providers, leading to increased open and direct competition with other providers. CSPs often offer differentiated services with specialized proprietary products and services. Standardization also reduces the efficacy of CSPs that use such differentiated service offerings to attract and maintain more clients.

For cloud collaboration to be viable in the current environment, researchers need to develop mechanisms that allow opportunistic collaboration among services without requiring standards and extensive changes to the cloud service delivery model. This approach will allow incremental provisioning of collaborative services to clients, which will continue to improve as more cloud services become interoperable in the future.

Cloud-based computing also introduces new security concerns that affect collaboration across multicloud applications, including the following:[8]

- increase in the attack surface due to system complexity,
- loss of client's control over resources and data due to asset migration,
- threats that target exposed interfaces due to data storage in public domains, and
- data privacy concerns due to multitenancy.

The research community is beginning to develop architectures, technologies, and standards to support collaboration among multiple cloud systems.

Some specific security issues associated with collaboration among heterogeneous clouds include

- establishing trust among different cloud providers to encourage collaboration;
- addressing policy heterogeneity among multiple clouds so that composite services will include effective monitoring of policy anomalies to minimize security breaches; and
- maintaining privacy of data and identity during collaboration.

Mechanisms for collaboration across multiple clouds must undergo a rigorous, in-depth security analysis to identify new threats and concerns resulting from collaboration. They must have the support of innovative, systematic, and usable mechanisms that provide effective security for data and applications. Such security mechanisms are essential for gaining the trust of the general public and organizations in adopting this new paradigm.

COLLABORATION FRAMEWORK FOR MULTICLOUD SYSTEMS

Our proposed framework for generic cloud collaboration allows clients and cloud applications to simultaneously use services from and route data among multiple clouds. This framework supports universal and dynamic collaboration in a multicloud system. It lets clients simultaneously use services from multiple clouds without prior business agreements among cloud providers, and without adopting common standards and specifications.

Use of proxies for collaboration

In the current environment, a client that wishes to simultaneously use services from multiple clouds must individually interact with each cloud service, gather intermediate results, process the collective data, and generate final results. The following restrictions in the current cloud computing model prevent direct collaboration among applications hosted by different clouds:

- *Heterogeneity and tight coupling.* Clouds implement proprietary interfaces for service access, configuration, and management as well as for interaction with other cloud components. Each service layer of a cloud tightly integrates with lower service layers or is highly dependent on the value-added proprietary solutions that the cloud offers. This heterogeneity and tight coupling prohibit interoperation between services from different clouds.

Proxies can facilitate collaboration without requiring prior agreements between the cloud service providers.

- *Preestablished business agreements.* The current business model requires preestablished agreements between CSPs before collaboration can occur. These agreements are necessary for clouds to establish their willingness to collaborate and establish trust with one another. The lack of such agreements prohibits multicloud collaborative efforts due to incompatible intentions, business rules, and policies. Moreover, collaborations resulting from preestablished agreements typically exhibit tight integration between the participants and cannot be extended to provide universal and dynamic collaboration.
- *Service delivery model.* Clouds use a service delivery model that provides service access to legitimate subscribing clients and denies all other requests because of security and privacy concerns. This prevents direct interaction between services from different clouds. Also, CSPs typically package their service offerings with other resources and services. This results in a tight dependency of a service on the hosting CSP. Such a service delivery model limits a client's ability to customize a service and use it in combination with service offerings from different CSPs.

A technique that could overcome these restrictions uses a network of proxies. A proxy is an edge-node-hosted software instance that a client or a CSP can delegate to carry out operations on its behalf. Depending on the context, the system can regard a network of proxies as a collection of virtual software instances connected via a virtual network or a set of physical nodes connected via an underlying network infrastructure.

The basic idea is to enable proxies that act on behalf of a subscribing client or a cloud to provide a diverse set of functionalities: cloud service interaction on behalf of a client, data processing using a rich set of operations, caching of intermediate results, and routing, among others. With these additional functionalities, proxies can act as mediators for collaboration among services on different clouds. Proxy deployment can be strategic—in close geographical proximity to the clouds, for example—to improve performance and facilitate execution of long-lived applications without additional user intervention.

As an example of proxy-facilitated collaboration between clouds, consider a case in which a client or CSP wishes to simultaneously use a collection of services that multiple clouds offer. First, the requesting entity chooses proxies to act on its behalf and to interact with cloud applications. A client or a CSP might employ multiple proxies to interact with multiple CSPs. It can select proxies based on, for example, latencies between proxies and clouds or workload conditions at various proxies.

Once it chooses proxies, the client or CSP delegates the necessary service-specific privileges to the proxies to carry out the service request using the necessary security precautions. These proxies can further delegate to other proxies if necessary and initiate the service request. In some instances, clients or CSPs can assign special roles to one or more proxies in the network to coordinate the operations in a service request among the multiple delegate proxies. Following delegation, the requesting entity need not further interact with the proxy network until the proxies complete the service request.

During execution of a service request, proxies would interact with cloud-based applications, playing the role of the service subscriber(s). By independently requesting services from the clouds, and by routing data between each other in a manner transparent to cloud applications, proxies can facilitate collaboration without requiring prior agreements between the CSPs. Proxies can also perform operations to help overcome incompatibilities among services to allow data exchange between them.

Architectural overview

Clouds consist of multiple network-connected resource clusters such as server farms, data warehouses, and so on that host geographically distributed virtual machines and storage components that ensure scalability, reliability, and high availability. A multicloud system that employs proxies for collaboration consists of three architectural components: multiple cloud computing systems, networks of proxies, and clients (or service users).

Such systems can use several possible strategies for placing proxies in the proxy network.

Cloud-hosted proxy. As Figure 1 shows, each CSP can host proxies within its cloud infrastructure, manage all proxies within its administrative domain, and handle service requests from clients that wish to use those proxies for collaboration. The proxy instances might need to be CSP-specific. For example, in Figure 1, both C1 and C2 might mutually and dynamically provision sharing and collaboration logic as proxy virtual instances within their respective administrative domains.

Proxy as a service. As Figure 2 shows, this scenario involves deploying proxies as an autonomous cloud that offers collaborative services to clients and CSPs. A group of CSPs that are willing to collaborate can manage this proxy-as-a-service cloud, or a third-party entity, a proxy service provider (PSP), can provide management. Clients directly subscribe to the proxy cloud service and employ them for intercloud collaboration.

Peer-to-peer proxy. Proxies can also interact in a peer-to-peer network managed by either a PSP or a group of CSPs that wish to collaborate. Another possibility is for proxies to have no collective management: each proxy in the peer-to-peer network is an independent entity that manages itself. In this case, the proxy itself must handle requests to use its services.

On-premise proxy. In the scenario shown in Figure 3, a client can host proxies within its organization's infrastructure (or on premises) and manage all proxies within its administrative domain. A client that wishes to use proxies for collaboration will employ its on-premises proxies, whereas CSPs that wish to collaborate with other CSPs must employ proxies that are within the domain of the service-requesting client.

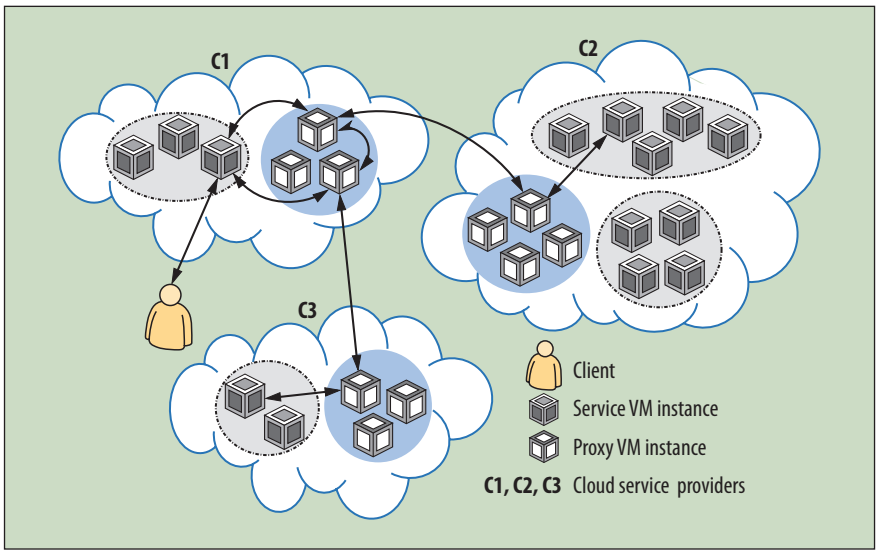


Figure 1. Client sends a request to cloud C1, which dynamically discovers the need to use services from clouds C2 and C3. C1 employs proxies to manage these interactions.

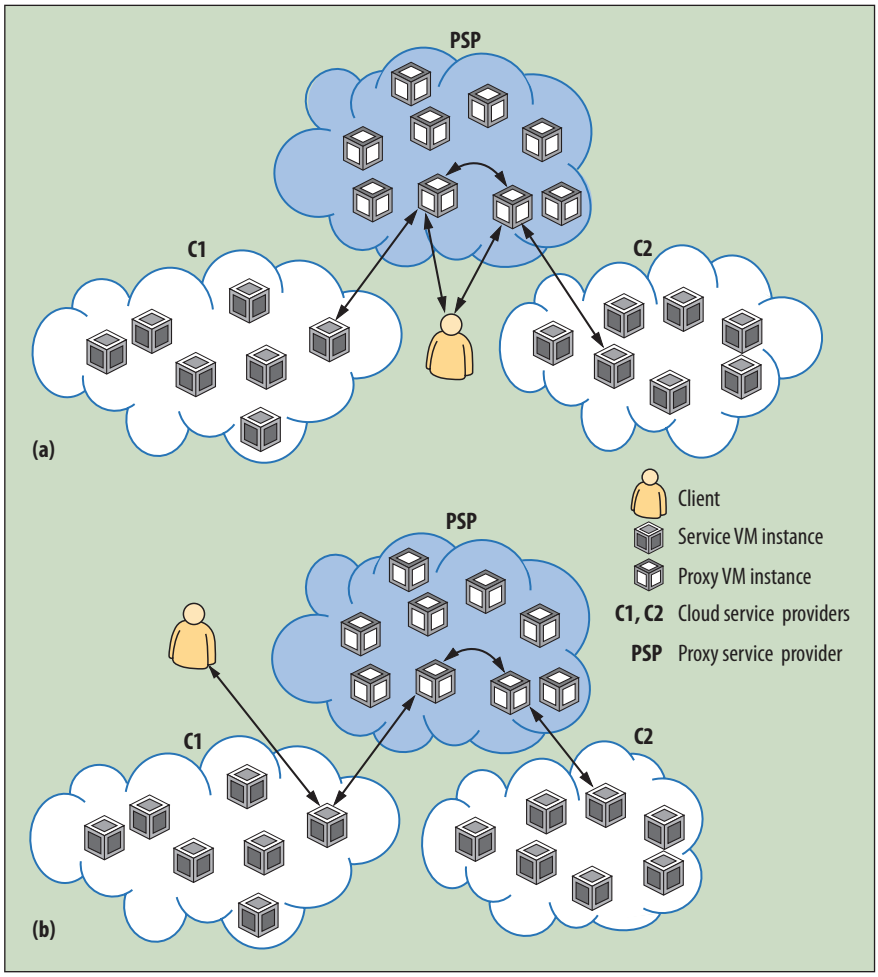


Figure 2. Proxy as a service. In this scenario, cloud service providers (CSPs) deploy proxies as an autonomous cloud system and offer it as a service to clients. (a) A client employs two proxies to interact with CSPs C1 and C2. (b) Alternatively, a client initiates a service request with C1, which then discovers the need for a service from C2. PSP: proxy service provider.

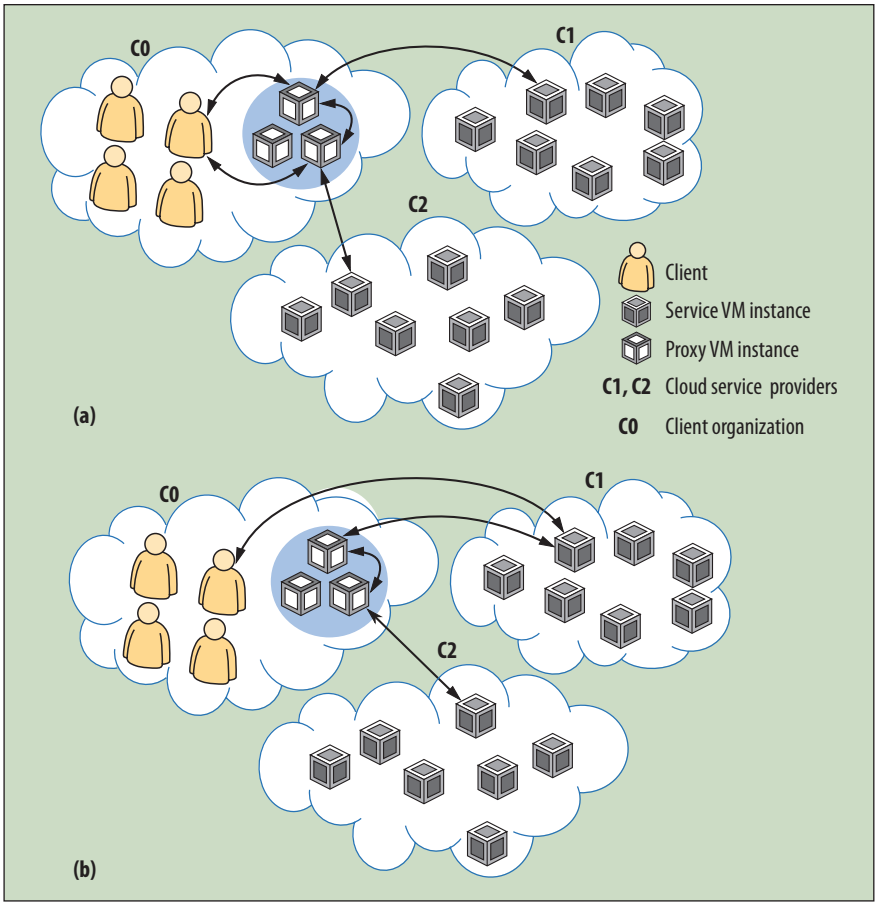


Figure 3. On-premises proxy. Clients deploy proxies within the infrastructure of their organization. (a) A client employs two proxies to interact with CSPs C1 and C2. (b) A client initiates a service request with C1, which then discovers the need for a service from C2.

Hybrid proxy infrastructure. A hybrid infrastructure can include on-premises, CSP- and PSP-maintained, and peer-to-peer proxies. Selecting proxies for collaboration will depend on the type of service being requested and the entity that initiates collaboration, among other factors. For example, clients that must initiate a service request with two CSPs can employ on-premises proxies for collaboration. On the other hand, a cloud-based application that discovers it needs a service from another CSP to fulfill a client's request can employ a CSP-maintained proxy.

The proposed architectures illustrate the various options that are available for deploying proxies to support collaboration. Developing these architectures serves as the first step in building a proxy-based, collaborative, multicloud computing environment.

A complete solution will entail several additional tasks. For example, an important task is a comprehensive study and evaluation of the proposed proxy-based architectures. Such an evaluation must cover each architecture's possible variations under diverse practical use cases and scenarios for multicloud collaboration. Based on this study, researchers can refine the proposed architectures, develop new variations to support different scenarios and use cases, and, if possible, merge the architectures into a universal proxy-based architecture for multicloud collaboration.

Another important task is developing a full suite of protocols and mechanisms that proxies must implement to support all the functionalities necessary for acting as mediators among services from multiple clouds. For example, supporting collaboration scenarios that migrate a client-subscribed virtual machine from one cloud to another requires techniques for translation between various virtual machine packages and distribution formats.

SECURITY ISSUES IN MULTICLOUD COLLABORATION

Researchers and industry specialists have highlighted several security issues in cloud computing, including isolation management, data exposure and confidentiality, virtual OS security, trust and compliance, and mission assurance.[8] Specific security issues emerge during dynamic sharing and collaboration across multiple clouds. In particular, issues pertaining to trust, policy, and privacy are a concern in multicloud computing environments.

Establishing trust and secure delegation

As in other IT systems, security in clouds relies heavily on establishing trust relationships among the involved entities. The need for trust arises because a client relinquishes direct control of its assets' security and privacy to a CSP. Doing so exposes a client's assets to new risks that are otherwise lessened or avoidable in an internal organization. These risks include insider security threats, weakening of data ownership rights, transitive trust issues with third-party providers in composite cloud services, and diminished oversight of system security.[8] A client must confer a high level of trust to a CSP with regard to its ability to implement effective controls and processes to secure assets. Thus, a client must be able to accept the higher levels of risk in using cloud-based services.

Using proxies moves the trust boundary one step further: clients and CSPs now must establish trust relationships with proxies, which includes accepting a proxy's security,

reliability, availability, and business continuity guarantees. Moreover, CSPs responding to service requests that a proxy makes on behalf of a client or another CSP must trust the proxy to legitimately act on behalf of the requesting entity. Establishing a trust relationship with proxies depends on the strategy used to establish, manage, and administer the proxy network. The entity managing the proxies must provide guarantees of its own trustworthy operation; additionally, it must provide assurances of the proxies' security, reliability, and availability.

From the client's point of view, employing on-premises proxies that are within the client's administrative domain can exacerbate trust issues. By using on-premises proxies, a client maintains control over its assets while proxies process them during a collaborative service request. Similarly, using proxies within the CSP's administrative domain lets the CSP exercise control over the proxies' operations, and thus it can trust the proxies to enable collaboration.

Proxy networks are a potential platform for developing proxy-based security architectures and solutions for multicloud systems. At a minimum, the proxy network must implement security and privacy mechanisms that mirror, extend, or complement similar mechanisms offered by clouds[8] to maintain asset protection outside the domain of clouds and client organizations. For example, to protect data at rest and data in transit, proxies must provide a trusted computing platform that prevents malicious software from taking control and compromising sensitive client and cloud application data. They must also guarantee data confidentiality and integrity during transmission through the proxy network, possibly by using standards such as the Transport Layer Security protocol.

In addition, clients, clouds, and proxies must implement mechanisms that ensure secure delegation, which entails the following:

- *On-the-fly agreements*. Delegating to a proxy must establish, on the fly, an explicit agreement between the delegator and proxy that lets the proxy act on the delegator's behalf. Techniques for delegation to a proxy must include mechanisms that restrict the proxy's behavior, including data and resource access, to comply with delegator-specified constraints.
- *Expected behavior*. After delegation, a proxy must not deviate from the expected behavior. It must act only on behalf of the delegator (a client or a CSP). After the proxy fulfills the service request, it can no longer act on the delegator's behalf. The proxy cannot modify the intended service request or misuse client assets, and it must not transitively delegate its capabilities to other proxies without the delegator's explicit consent.

The technologies for secure delegation to proxies include the use of warrant-based proxy signatures for delegation of signing rights to provide authentication of proxies.[9] Simple public-key infrastructure authorization certificates[10] or the OAuth protocol[11] can facilitate the secure delegation of access rights and permissions. Researchers must thoroughly evaluate existing secure-delegation techniques and develop a comprehensive protocol suite that implements mechanisms that support secure delegation and proxy operation.

Policy heterogeneity and conflicts

When proxies enable dynamic collaboration between multiple CSPs, heterogeneous security policies can be the source of policy conflicts that result in security breaches. Proxies must monitor for and defend against such breaches. Even though existing policy evaluation mechanisms can verify individual domain policies, security violations can easily occur during integration.[12] In multicloud collaborations using proxies, service requirements can drive dynamic, transient, and intensive interactions among different administrative domains. Thus, a proxy's policy integration tasks must address challenges such as semantic heterogeneity, secure interoperability, and policy evolution management. The design of access control policies for multicloud collaboration must permit careful management by proxies while ensuring that policy integration does not lead to security breaches.

To protect data at rest and data in transit, proxies must provide a trusted computing platform that prevents malicious software from taking control and compromising sensitive client and cloud application data.

Policy analysis generally includes property verification and conflict detection, as well as an analysis of the differences between policy versions. For a collaborative service, a proxy must deal with several registered services from multiple clouds as well as proxies. This requires various proxies to locally conduct policy integration and decomposition.

Policy integration aims to generate agreement on access rights for each party involved in a collaborative project. A policy integration process for intercloud collaboration must systematically handle potential conflicts and resolution problems. Proxies must analyze relationships between policies to detect and resolve policy anomalies using mechanisms that easily adapt to handle composite policies evaluated as a whole. Possible policy anomalies include policy inconsistencies and inefficiencies.

Policy inconsistency. Access control policies reflect security requirements, which should be consistent within

and across multiple participating parties to accommodate the dynamic and complex nature of multicloud environments. Policy inconsistencies can result in security and availability problems; they include the following:

- *Contradiction*. Two policies are contradictory if they have different effects on the same subjects, targets, and conditions. Contradictions are the most common form of policy conflicts.
- *Exception*. A policy is an exception of another policy if they have different effects, but one policy is a subset of the other. The exception might not be a policy conflict, but access policy evaluation mechanisms commonly use exceptions to exclude a specific access request from a general access permission.
- *Correlation*. Two policies are correlated if they have different effects but intersect each other. In this case, one policy permits the intersection, but the other does not. This is a partial policy conflict.

Policy inefficiency. The composition of policies from multiple origins can result in a large collection of policies controlling the access to federated applications in multiclouds. Since an access request's response time largely depends on the number of policies that proxies must parse, inefficiencies in composite policies can adversely affect performance. Inefficiencies in composite policies include

- *redundancy*—a policy is redundant if every access request that matches the policy also matches another policy with the same effect; and
- *verbosity*—similar to data element merging in data integration, policy composition can merge similar policies from different origins; resolving the policy verbosity during composition affects the policy size.

With cloud computing initiatives, the scope of insider threats, a major source of data theft and privacy breaches, is no longer limited to the organizational perimeter.

Once proxies identify conflicts, they must use conflict resolution strategies to resolve them. However, current conflict resolution mechanisms have limitations. For example, existing conflict resolution mechanisms—including Extensible Access Control Markup Language (XACML) policies—are too restrictive because they only allow the selection of one resolution algorithm to resolve all the identified conflicts.

Multicloud environments require adaptively applying different algorithms to resolve different conflicts. It is therefore necessary to develop a flexible and extensible conflict resolution approach to achieve fine-grained conflict resolution. Such an approach must let a proxy automatically map different conflict resolution strategies that resolve different conflicts.

Situations in which a policy component becomes involved in multiple conflicts also require a correlation mechanism that identifies dependent relationships among conflicting segments. Such a mechanism ensures that conflict resolution does not introduce new policy violations during the resolution process. Earlier research applied an approach for detecting and resolving policy anomalies to healthcare domains.[13,14] Specifically, our preliminary study demonstrates how to achieve compliance and conflict analysis in EMR management systems, as applied to data sharing and Health Insurance Portability and Accountability Act (HIPAA) policies.

Identity attributes and data privacy

In shared computing environments like clouds, protecting the privacy of client assets is critical.[8] The privacy issues pertaining to both data and identity.

Identity attributes privacy. Data as a service (DaaS), such as Amazon S3 and Microsoft Azure, is an emerging cloud service in which organizations can seamlessly store data in the cloud and retrieve it based on access control policies that cover legal requirements and organizational policies. An expressive access control model, such as XACML, can specify access control policies on protected objects in terms of a subject's properties, called *identity attributes*. These can include a subject's email address, organizational role, age, and location of access. Such an *attribute-based access control* (ABAC) model provides fine-grained data access and expresses policies closer to organizational policies.

A crucial issue in this context is that identity attributes required by subjects to access protected objects often encode sensitive information. Many existing cloud data services provide similar access control models, in which individual and organizational privacy, a key requirement for digital identity management, is unprotected.

Also, with cloud computing initiatives, the scope of insider threats, a major source of data theft and privacy breaches, is no longer limited to the organizational perimeter. Multicloud environments exacerbate these issues because proxies can access data (which the environment might dynamically move or partition across different clouds) on behalf of clients. Revealing sensitive information in identity attributes to proxies that grants them authorization to access the data on behalf of clients is not an attractive solution. Thus, assuring the private and consistent management of information relevant to ABAC becomes more complex in multicloud systems.

In multicloud environments, where proxies use ABAC to retrieve client data from the clouds, clients need to

hide their identity attributes from both proxies and CSPs to preserve the privacy of sensitive client information. However, clients must still give proxies the information that grants them access to requested data. This requirement calls for the use of identity attribute and data encoding techniques that, used together, permit oblivious data transfer between CSPs, proxies, and clients while providing privacy-preserving ABAC.

The techniques for encoding client identity attributes must permit clients to transfer the encoded attributes to proxies; the proxies, in turn, must convince CSPs of the ownership and validity of the encoding, without having the client reveal its identity attributes to either entity. Data and identity attribute encoding techniques must ensure that decoding the data is possible when the identity attributes match the ABAC policies, without revealing the attribute to the proxy or the CSP.

Client data privacy. Often, clients must protect data privacy before sharing the data.

Consider an example in which multiple medical insurance companies, each of which has a designated CSP, would like to share customer data to have a much larger customer database from which to obtain useful statistical query results. One CSP might have an application that requires information on the percentage of male construction workers in the US who are younger than 40 and have respiratory diseases. This would require collecting data from multiple CSPs for the analytical results to be meaningful, since the data from one CSP might be inadequate (after filtering for multiple selective predicates) or atypical (say, one CSP only has data for customers in a particular region of the US).

In this example, the disease attribute of records is sensitive and requires protection when shared among multiple CSPs. Using encryption is not a viable option because maintaining the data's utility is a key requirement for many applications. Most applications require a well-balanced tradeoff between formal privacy and practical utility.

Privacy protection methods (other than encryption) fall broadly into two categories:[15]

- *data perturbation* (also known as input perturbation), which adds some form of noise to the data itself, and
- *output perturbation*, which adds noise to the otherwise accurate query answers.

Earlier research studied data privacy in outsourcing data aggregation services.[16] Regardless of the methods used to maintain data privacy, the resulting solution must scale to use for large amounts of data and many CSPs.

Perturbation methods often produce data with high redundancies, which can lead to scalability issues in multicloud environments; a client's request for data can result in a large communication overhead in the proxy network. Compression methods such as dictionary encoding can reduce both communication and query processing costs—for example, CSPs and proxies can perform much of the query processing over the encoded format.[17]

Favorable solutions to ensure data privacy must employ flexible data perturbation methods that provide control over the tradeoff between the privacy guarantee and the utility of the query results. Perturbation methods must provide high accuracy for queries that involve a large number of records. At the same time, they must introduce large amounts of noise in the results for queries over a few records, which is desirable for privacy.

In the multiple-CSP context, a CSP can use local data perturbation techniques to perturb its sensitive data and then ship it to another CSP for collaborative query processing. Local techniques permit query processing at one site to avoid on-the-fly data communication costs. Moreover, when a query itself must be private, a CSP can limit query processing to its own site by using local techniques.

Favorable solutions to ensure data privacy must employ flexible data perturbation methods that provide control over the tradeoff between the privacy guarantee and the utility of the query results.

In some applications, the receiving CSP need not perturb its own sensitive data. These situations present opportunities to further optimize the accuracy and efficiency of query processing that researchers can explore by judiciously determining which CSP should answer a particular query (when queries are not private and sharable). Finally, multicloud scenarios require new privacy definitions that will allow formal proofs of privacy guarantees for protection schemes.

To facilitate dynamic collaboration between clouds, we proposed a framework that uses proxies to act as mediators between applications in multiple clouds that must share data. Our proposed framework has the potential to overcome several restrictions in the current cloud computing model that can prevent dynamic collaboration among applications hosted by different cloud systems.

Future research directions for the proposed framework include refining the proxy deployment scenarios and development of infrastructural and operational components of a multicloud system. This must be accompanied by implementation of an experimental platform using open source tools and libraries that work in combination with real-world cloud services to evaluate the

system's functionality and limitations, and make further refinements.

Currently, our research team is working toward a single viable proxy deployment strategy based on use cases, trust, and security requirements. We are also developing specifications to instantiate, deploy, maintain, and release proxy virtual machines reliably and securely, along with a suite of proxy services to support various collaboration use cases. Our incremental approach to the development of proxy services for collaboration initially provides support for simple use cases, later progressing to more complex use cases.

References

1. P. Mell and T. Grance, *The NIST Definition of Cloud Computing*, special publication 800-145, Nat'l Inst. Standards and Technology, 2011, p. iii + 3.
2. D. Bernstein and D. Vij, "Intercloud Security Considerations," *Proc. 2nd Int'l Conf. Cloud Computing* (CloudCom 10), IEEE Press, 2010, pp. 537-544.
3. R. Buyya et al., "Market-Oriented Cloud Computing: Vision, Hype, and Reality of Delivering Computing as the 5th Utility," *Proc. 9th IEEE/ACM Int'l Symp. Cluster Computing and the Grid* (CCGRID 09), IEEE CS, 2009, pp. 599-616.
4. B. Rochwerger et al., "Reservoir—When One Cloud Is Not Enough," *Computer*, Mar. 2011, pp. 44-51.
5. M.P. Papazoglou and W. van den Heuvel, "Blueprinting the Cloud," *IEEE Internet Computing*, Nov./Dec 2011, pp. 74-79.
6. S. Ortiz Jr., "The Problem with Cloud Computing Standardization," *Computer*, July 2011, pp. 13-16.
7. P. Mell and T. Grance, "Perspectives on Cloud Computing and Standards, NIST Information Technology Laboratory," Nat'l Inst. Standards and Technology, 2008; http://csrc.nist.gov/groups/SMA/ispab/documents/minutes/2008-12/cloud-computing-standards_ISPAB-Dec2008_P-Mell.pdf.
8. W. Jansen and T. Grance, *Guidelines on Security and Privacy in Public Cloud Computing*, special publication 800-144, Nat'l Inst. Standards and Technology, 2011, p. x + 70.
9. S. Chandrasekhar et al., "Efficient Proxy Signatures Based on Trapdoor Hash Functions," *IET Information Security*, Dec. 2010, pp. 322-332.
10. C.M. Ellison et al., *SPKI Certificate Theory*, IETF RFC 2693, Sept. 1999; www.ietf.org/rfc/rfc2693.txt.
11. E. Hammer-Lahav, ed., *The OAuth 1.0 Protocol*, IETF RFC 5849, Apr. 2010; http://tools.ietf.org/html/rfc5849.
12. Y. Zhang and J.B.D. Joshi, "Access Control and Trust Management for Emerging Multidomain Environments," *Ann. Emerging Research in Information Assurance, Security and Privacy Services*, Emerald Group Publishing, 2009, pp. 421-452.
13. J. Jin et al., "Patient-Centric Authorization Framework for Electronic Healthcare Services," *Computers & Security*, Mar.-May 2011, pp. 116-127.
14. R. Wu, G.J. Ahn, and H. Hu, "Towards HIPAA-Compliant Healthcare Systems," *Proc. 2nd ACM Int'l Symp. Health Informatics* (IHI 12), ACM, 2012, pp. 593-602.
15. N.R. Adam and J.C. Wortmann, "Security-Control Methods for Statistical Databases: A Comparative Study," *ACM Computing Surveys*, Mar. 1989, pp. 515-556.
16. L. Xiong, S. Chitti, and L. Liu, "Preserving Data Privacy in Outsourcing Data Aggregation Services," *ACM Trans. Internet Technology*, Aug. 2007, p. 17.
17. D.J. Abadi, S. Madden, and M. Ferreira, "Integrating Compression and Execution in Column-Oriented Database Systems," *Proc. ACM SIGMOD Int'l Conf. Management of Data* (SIGMOD 06), ACM, 2006, pp. 671-682.

Mukesh Singhal *is a Chancellor's Professor in the Computer Science and Engineering program at the University of California, Merced. His research interests include distributed computing, cloud computing, security, and networks. Singhal received a PhD in computer science from the University of Maryland, College Park. He is an IEEE Fellow. Contact him at msinghal@ucmerced.edu.*

Santosh Chandrasekhar *is a postdoctoral research scholar at the University of California, Merced. His research interests include building cryptographic protocols for securing wired and wireless networks and their applications. Chandrasekhar received a PhD in computer science from the University of Kentucky. Contact him at schandrasekhar@ucmerced.edu.*

Tingjian Ge *is an assistant professor of computer science at the University of Massachusetts Lowell. His research interests include database security and privacy, uncertain and probabilistic data, and scientific data management. Ge received a PhD in computer science from Brown University. Contact him at ge@cs.uml.edu.*

Ravi Sandhu *is the executive director of the Institute for Cyber Security at the University of Texas at San Antonio, where he holds the Lutcher Brown Endowed Chair in Cyber Security. His research interests focus on cybersecurity practice and education. Sandhu received a PhD in computer science from Rutgers University. He is an IEEE, ACM, and AAAS Fellow. Contact him at ravi.sandhu@utsa.edu.*

Ram Krishnan *is an assistant professor in the Department of Electrical and Computer Engineering at the University of Texas at San Antonio. His research interests include the authorization aspects of computer security. Krishnan received a PhD in computer science from George Mason University. Contact him at ram.krishnan@utsa.edu.*

Gail-Joon Ahn *is an associate professor in the School of Computing, Informatics, and Decision Systems Engineering at Arizona State University, and the director of security engineering for the Future Computing Laboratory. His research interests focus on vulnerability and risk management, access control, and security architectures for distributed systems. Ahn received a PhD in computer science from George Mason University. Contact him at gahn@asu.edu.*

Elisa Bertino *is a professor of computer science at Purdue University, research director for the Center of Education and Research in Information Assurance and Security (CERIAS), and director of the Discovery Park Cyber Center. Her research interests focus on data privacy and security. Bertino received a PhD in computer science from the University of Pisa, Italy. She is an IEEE and ACM Fellow. Contact her at bertino@cs.purdue.edu.*