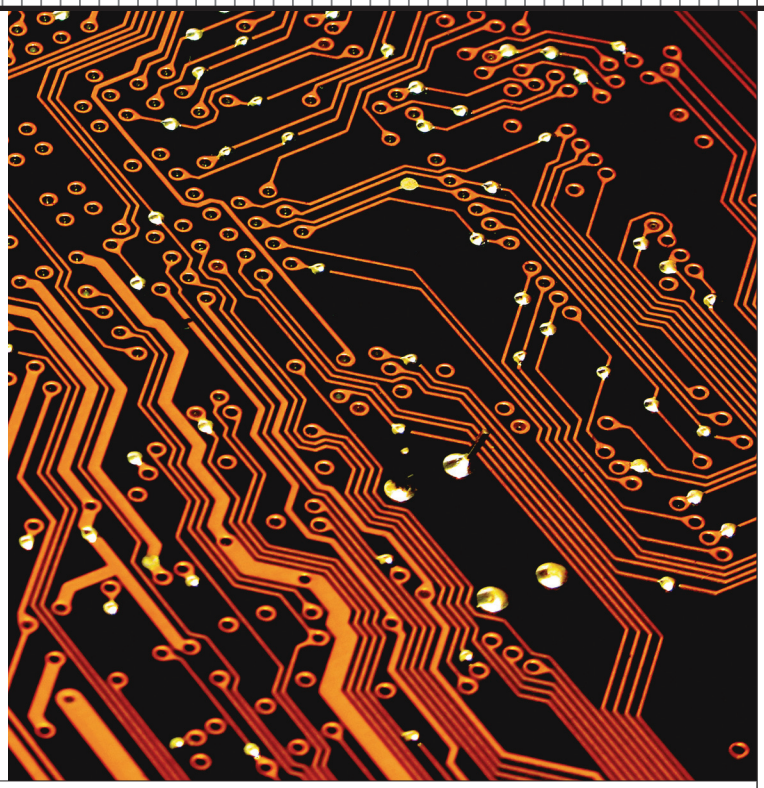
On-Demand Security Architecture for Cloud Computing



Jianyong Chen, Yang Wang, and Xiaomin Wang

Shenzhen University, China

An architecture that differentiates security according to service-specific characteristics avoids an unnecessary drain on IT resources by protecting a variety of cloud computing services at just the right level.

Cloud computing makes it possible for content providers to quickly deploy and scale services and benefit from low-cost, pay-by-use models, while service users enjoy the flexibility that Internet-based computing provides. Cloud services generally take the form of software as a service (SaaS), platform as a service (PaaS), or infrastructure as a service (IaaS). Successful commercial solutions include Amazon's EC2/S3, Google Apps, and force.com.

However, the very flexibility and rapid provisioning that cloud computing offers pose serious obstacles to any security architecture.[1] Users find it difficult to fully trust cloud-based services because cloud-based data storage and protection methods are largely user transparent. There is no way to know, for example, if the service providers have properly deleted users' purged data or if they are saving it for their own reasons, such as passing on the user's name to third parties offering products related to the provided service or extracting privacy information for malicious use.[2]

Current research on cloud security is still in the early stages, and no universal model or set of techniques has yet emerged. Methods include segregating user resources during data processing to prevent widespread virus infection,[2] the use of a third-party auditor to verify the integrity of data stored in the cloud,[3,4] and access control based on data attributes and semantics.[5,6]

Some trust management experts recommend applying multiple security policies to authenticate users, manage identities, and protect data from unauthorized users. Amazon administrators, for example, log and routinely audit any access to customers' data or operating systems.[6]

Each of these research efforts aims to develop a security solution for a specific threat, yet such methods are incompatible with cloud services, which sometimes have vastly different security requirements. Some services involve public information that needs only basic security. Others, such as banking transactions, involve more sensitive information. To date, no single security architecture satisfies this requirements mix. As the "Why Not Protect at the Highest Level?" sidebar describes, the one-level-fits-all approach of traditional client-server architectures wastes resources and makes service use unnecessarily complex.

To fill the need for a more discerning security architecture, we are exploring a security-on-demand design that applies security algorithms and protocols according to three stages in the service data's life cycle: in transmission,

0018-9162/12/$31.00 © 2012 IEEE Published by the IEEE Computer Society

WHY NOT PROTECT AT THE HIGHEST LEVEL?

Although strong security is important in many services, such as e-commerce and telemedicine, many other services, such as the provision of public information, can function with much less security. Even users of the same service can have different security needs because their data might not have the same assets. For example, a user needs a high security level for a voice service that concerns a business discussion, but only a low level for the same service when calling a friend to meet for lunch. Similarly, an e-mail service user needs content encryption when the message contains sensitive information, but only plaintext for general e-mail.

Client-server systems tend to use the strongest security solution to protect all network services, but such an approach is not effective for cloud computing, the main advantages of which are ease of service use and IT resource savings. The stronger the security, the greater the consumption of computing, memory, and bandwidth resources and the more difficult the service is to use, requiring manual configuration of security mechanism parameters. Thus, protecting services and data at a higher level than they need erodes the advantages of a cloud-computing platform.

When security strength—what it takes to break the security mechanism—increases, users must rely on more complex operations to order and use a service. A strong security authentication, for example, requires not only a password but also a smart card and sometimes a fingerprint. Although such multifactor authentication is harder to compromise than single-factor methods, it greatly increases the complexity of the authentication process for the user and thus makes the service less attractive. Other methods to increase authentication strength, such as longer passwords, more complex password composition, or more frequent password changes, also put a burden on the user.

Consequently, using the strongest security for all cloud computing services is not practical because convenience is a major reason people want these services. On-demand security is a much better fit because it automatically differentiates security strength according to service type, the security level that users specify, and access network risk. This approach provides enough security for the least amount of IT resource consumption and preserves the service's ease of use.

in process, or in storage. The architecture matches the requirement to one of these three data stages, ensuring the least IT resource consumption per service and adjusting the service's ease of use accordingly.

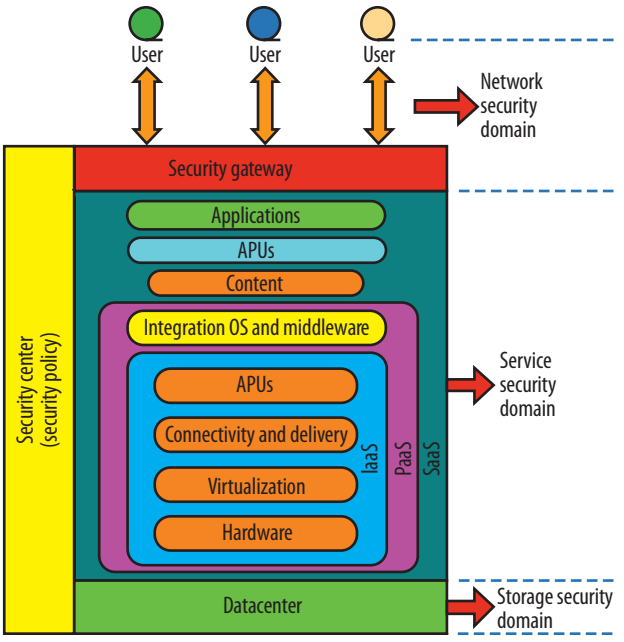


Figure 1. Three security domains in the on-demand cloud computing architecture. The security center manages cloud computing according to the security policy. The datacenter stores users' data. The security gateway manages all communications between the network and the service security domains, thus acting as a detection point if an attack occurs.

DEFINING SECURITY DOMAINS

Dividing the cloud service and Internet transmission into several security domains with each domain governed by its respective security policy can simplify the deployment of solutions to cloud service security.

As Figure 1 shows, our architecture is based on the network, service, and storage security domains, which reflect the three service data stages.

Figure 2 shows how the three domains interrelate to ensure that a service is protected in all three stages.

Network

The main threats while data is in transmission are fabricated identity, man-in-the-middle, and denial-of-service attacks. To protect against these threats, the network security domain includes mechanisms such as the Secure Socket Layer/Transport Layer Security (SSL/TLS) protocols, IPSec, network-based intrusion detection, and traffic cleaning.

The security gateway, which mediates all communications to and from the system, is an important entity in this domain because it enables more fine-grained access control. If a malicious act occurs, such as a distributed denial-of-service attack,[7] the gateway can immediately limit or even turn off malicious communication, thus thwarting the attacker. For legitimate connections, the network security domain specifies using a security protocol such as SSL or IPSec to protect against possible man-in-the-middle attacks and information leaks.

Service

The main threats to data in the cloud services (IaaS, PaaS, and SaaS) are a fabricated service process, an illegally

controlled service, and malicious service interruption. To address these threats, the service security domain includes mechanisms such as authentication, authorization, vulnerability scanning, data isolation, and virus detection. To protect legitimate services from illegal control and process interruption, an intrusion detection and prevention system monitors all user actions.

The system can also use honeypot technology—a trap set to detect, deflect, or in some manner counteract attempts at unauthorized use—to capture malicious actions at intervals.[8] To avoid viral infection and service hijacking, SaaS, PaaS, and IaaS for each user can run in logical isolation.

Storage

The main threats while data is in storage are unauthorized access and data alteration and theft. Protection mechanisms include encryption, marking data with different access levels to enable access control, and integrity verification. Backup techniques, such as a redundant array of independent disks and data recovery, insure against data loss.

ARCHITECTURAL COMPONENTS

Figure 3 shows the three layers of our on-demand security architecture. The input layer receives the user-specified security level, access network risk, and service type. The policy layer determines parameters for security mechanisms in every security domain according to the three inputs. The security mechanisms layer protects a specific service according to the security parameters from the policy layer.

Input layer

The three inputs into the input layer determine which security policy will govern the service.

Security level. The service provider's system must permit authorized users simultaneous access according to security clearance and authorization level and keep unauthorized users out. Because the application environment poses a certain risk to the system's ability to perform these tasks, the security level must reflect both what a specific service requires and the risk to the system in providing that service securely.

Each service provider offers a minimum service security level, which means that users can choose not to set a security level and still receive minimum protection.

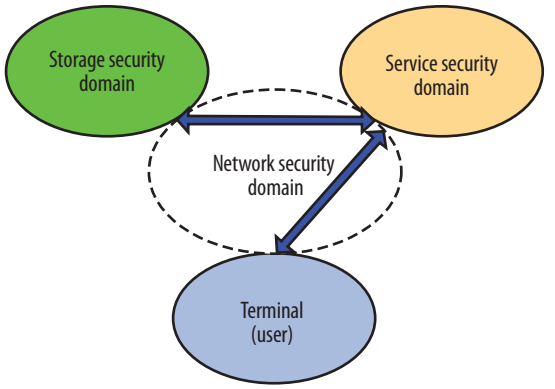


Figure 2. Interrelationship of security domains. The network security domain is a bridge for the service and storage security domains and for the user and the service security domain, thereby ensuring data protection from origination to storage.

The security level is not the same as security strength. Security level refers to the difficulty of breaking into a system and reflects both security strength and risk to the application environment. Security strength, on the other hand, reflects only the difficulty of breaking the security mechanism.

Traditional security planning has maintained the security level in high-risk system environments by increasing the strength of security mechanisms. Our architecture adjusts security strength according to the specific service needs as well as the risk.

Service type. Our architecture includes service type in the input layer because different service types require

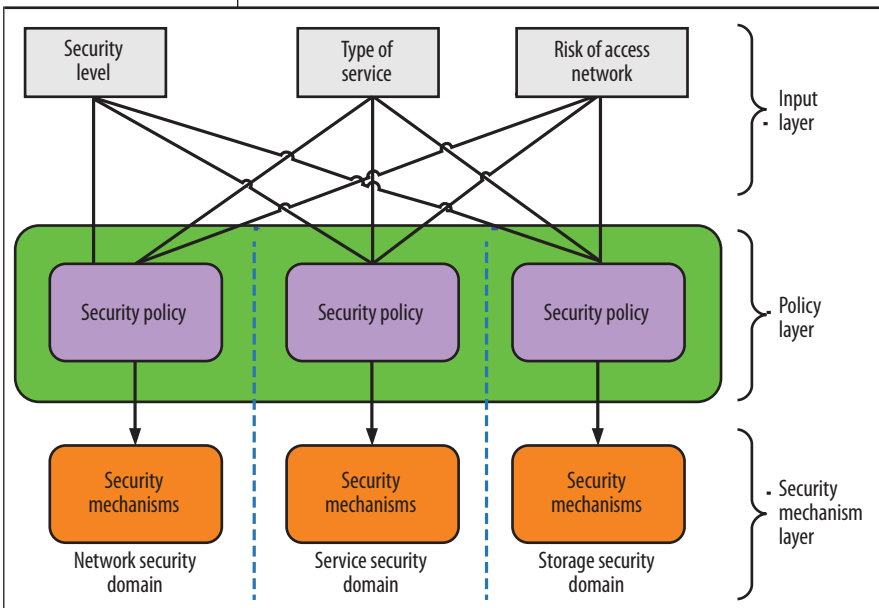


Figure 3. On-demand security architecture. The architecture is divided into input, policy, and security mechanism layers. The input layer receives the security level, service type, and access risk, which feed into the policy layer to determine which security policy will govern the service through the security mechanisms in the bottom layer.

different security mechanism combinations. A multimedia service, for example, is sensitive to time delay, allows a certain degree of packet loss, and does not require integrity verification. For a file transmission service, in contrast, integrity verification is a crucial protection mechanism.

Users need not specify service type. Once a user starts a specific service, the cloud service automatically configures the service type input.

Access network risk. The risk of attack while the service passes through the access network—such as 3G, public Wi-Fi, or wired office networks—depends on the network being used. The risk is relatively high with a public Wi-Fi access network and relatively low with a wired intranet.

Users need not specify the access network risk. The cloud service can acquire that value from the terminal location, the IP address range at the user's terminal, or border entities at the access network. Normally, the higher the risk, the stronger the security mechanisms must be.

Specifying a service's security level is a simple selection process on a familiar computing interface, such as a webpage, requiring no special security knowledge.

Policy layer

In the policy layer, three security policies receive inputs simultaneously and produce the security mechanism parameters on the basis of the specified security level, service type, and access network risk. Because the three inputs decide the strength and combination of security mechanisms, the security policy's main role is to evaluate those inputs and produce the appropriate mix of security parameters. These parameters, in turn, ensure that security mechanisms protect the service at a consistent security level.

Each security policy produces the parameters that will activate security mechanisms in one of the three domains. In the network security domain, for example, IPSec is an important security mechanism. The Security Association (SA) handles many of IPsec's security parameters, such as protocol type, package mode, encryption algorithm, and key life cycle.

To protect data in the network domain, our architecture's security policy produces the needed SA security parameters for that service. From that point, the SA security parameters drive the IPsec to protect data flow.

Security mechanism layer

Each domain is governed by a particular security policy, which in turn provides the appropriate security mechanisms, such as IPsec in the network security domain, honeypot in service security domain, and data encryption/decryption in the storage security domain.

Some security mechanisms are appropriate for more than one security domain and take different names, depending on their function. For example, intrusion detection is network-based in the network security domain, but it becomes host-based in the service security domain. The antivirus mechanism is also appropriate for both the network and service security domains.

IMPLEMENTATION ASPECTS

One traditional implementation concern is how users upgrade the security level beyond the service provider's default. Unlike many methods, which require manual configuration, our architecture makes the upgrading process extremely user-friendly. Specifying a service's security level is a simple selection process on a familiar computing interface, such as a webpage, requiring no special security knowledge. A user who believes that the service will handle sensitive personal information can simply choose a high security level before using the service; the architecture does the rest of the configuring.

Another implementation issue is how to accommodate different user platforms. In our architecture, end-to-end security mechanisms for differentiated security are based only on the user's browser. The security mechanisms are independent of the user's hardware and operating system.

Finally, although each security domain has its own security policy, the domains strongly correlate because each service has a service ID, which the domains share. The security policy in every domain receives the same inputs simultaneously, and once a user orders a service, data flow for the same service is embedded in its ID across domains. Using the service ID as a reference, any of the three domains can protect a specific service's data flow at the needed security level.

APPLICATION SCENARIO

Figure 4 shows a sample application of our architecture, in which Alice initiates a videoconference from her hotel room with Bob and George at the office, specifying a high security level for her communication.

Cloud computing automatically configures service type and access network risk. To protect the videoconference, mechanisms provide authentication in the service security domain and confidentiality assurance in the network security domain.

This scenario involves two uses of the network security domain: from the hotel (Alice) to the service provider and from the office (Bob and George) to the service provider. The service and storage security domains are the service provider's responsibility.

All three inputs—security level, service type, and access network risk—require configuration. Alice specified a high

security level, and the cloud service automatically configures the appropriate values for service type and access network risk. The service type for a videoconference must be real-time, and the access network risk is high on the hotel side and low on the office side, reflecting different security strengths even though the specified security level is high.

According to these inputs, the security policy for the network security domain (hotel side) produces parameters to drive IPsec to protect data flow between Alice and the service provider. For the office side, the security strength (not security level) is low, so the data flow from the office to the service provider is in plaintext.

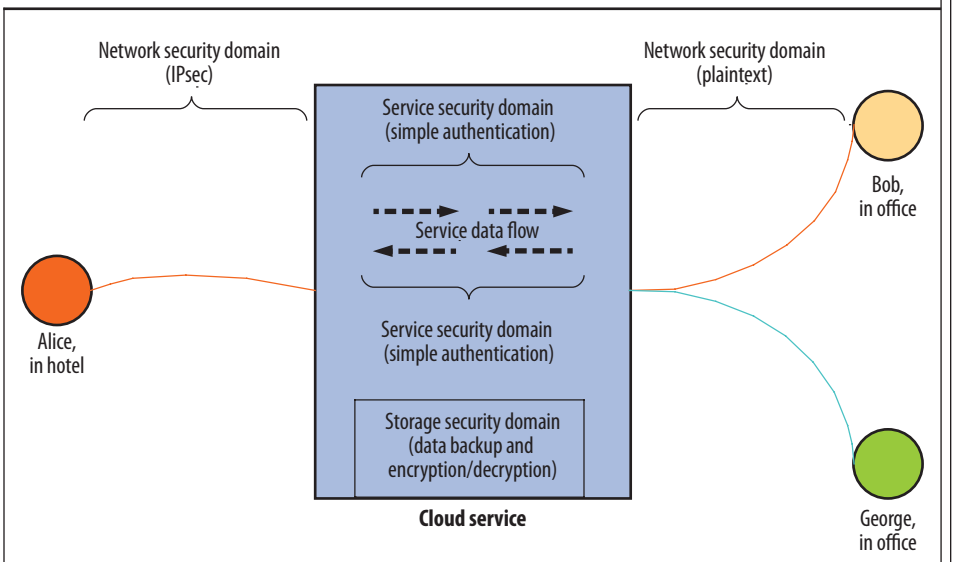


Figure 4. On-demand security for cloud-based videoconference service from Alice to Bob and George at the office. The application shows how the architecture adjusts security mechanisms according to need. Under the same security level (high), the architecture provides strong security from Alice to the service provider, but weaker security for data from the office to the service provider because the required protection is lower.

Similarly, in the service security domain, the security policy initiates multifactor authentication to authenticate Alice but only simple authentication to authenticate Bob and George. In the storage security domain, data flow can be encrypted and stored.

In this application, the security strength for the same service is different across access networks, yet the specified security level remains constant. The security policy controls all these configurations automatically.

On-demand security preserves the benefits of cloud computing by saving IT resources and not burdening the service user with tedious security specifications, such as configuring security mechanism parameters.

Our architecture offers several advantages. Each security domain faces different security threats but draws from the same set of security mechanisms to address those threats. Consequently, each domain can focus on its own issue according to the dictates of its tailored security policy. Dividing the larger security universe into three specific domains simplifies security policy delegation and makes it more practical.

In cloud computing, the same provider might not offer network management, service provision, and storage. Thus, having three domains also fits well with different providers for these functions.

Another advantage is simplicity. The user platform needs to configure only three inputs. Once the users order a service and specify the security level, the platform automatically factors in the service type and access network risk. Consequently, inputs are easy to manage and configure.

Our architecture also makes it more practical to evolve traditional network systems to cloud computing. Because the security policy provides security on demand, there is no need to adapt security mechanisms for every domain. Consequently, architects can add security mechanisms in an existing network to our architecture without fundamentally changing them. Using existing network resources represents a substantial savings in efforts to deploy cloud computing.

Acknowledgments

The work described in this article was supported by the China-Finland Cooperation Project on the Development and Demonstration of Intelligent Design Platform Driven by Living Lab Methodology (grant 2010DFA12780), Shenzhen Fundamental Research Plan (grant JC201005250045A), and National Natural Science Foundation of China (grants 61170283 and 61171072).

References

1. H. Takabi, J.B.D. Joshi, and G.J. Ahn, "Security and Privacy Challenges in Cloud Computing Environments," *Computer*, June 2010, pp. 24-31.
2. S. Subashini and V. Kavitha, "A Survey on Security Issues in Service Delivery Models of Cloud Computing," *J. Network and Computer Applications*, vol. 34, no. 1, 2011, pp. 1-11.
3. C. Wang et al., "Toward Publicly Auditable Secure Cloud Data Storage Services," *IEEE Network*, vol. 24, no. 4, 2010, pp. 19-24.

4. Q. Wang et al., "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," *IEEE Trans. Parallel and Distributed Systems,* vol. 22, no. 5, 2011, pp. 847-859.
5. L.K. Hu, S. Yi, and X.Y. Jia, "A Semantics-Based Approach for Cross Domain Access Control," *J. Internet Technology*, vol. 11, no. 2, 2010, pp. 279-288.
6. G. Pallis, "Cloud Computing: The New Frontier of Internet Computing," *IEEE Internet Computing*, vol. 14, no. 5, 2010, pp. 70-73.
7. R.P. Lua and K.C. Yow, "Mitigating DDoS Attacks with Transparent and Intelligent Fast-Flux Swarm Network," *IEEE Network*, vol. 25, no. 4, 2011, pp. 28-33.
8. V.H. Pham and M. Dacier, "Honeypot Trace Forensics: The Observation Viewpoint Matters," *Future Generation Computer System—Int'l J. Grid Computing and E-science*, vol. 27, no. 5, 2011, pp. 539-546.

Jianyong Chen *is a professor in the Computer Science and Technology Department at Shenzhen University, China. His research interests include network security and artificial intelligence. He received a PhD in information technology from the City University of Hong Kong. He is a member of IEEE. Contact him at jychen@szu.edu.cn.*

Yang Wang *is a graduate student at Shenzhen University, China. His research interests include network security. Wang received a BSc in computer science from Shenzhen University. Contact him at wangyangwla@sina.com.*

Xiaomin Wang *is an associate professor in the Computer Science and Technology Department at Shenzhen University, China. His research interests include network security and Internet applications. Wang received an MSc in computer science from Xidian University, China. Contact him at wangxm@szu.edu.cn.*