Adopting Trusted Third Party services for Multi-level Authentication accessing cloud

Vivekananth.P (1st) Dr.Ritish Khanna (2nd)
[1] Research Scholar
Department of Computer Science
CMJ University Shillong Meghalaya
[2] Faculty Isara institute of management and professional studies Delhi

Abstract*:*
Cloud computing is an emerging, on-demand and internet-based technology. It provides variety of services over internet such as, software, hardware, data storage and infrastructure. This technology has been used by worldwide customers to improve their business performance. At present, authentication is done in several ways: such as, textual, graphical, bio-metric, 3D password and third party authentication. The framework have seven processes, including: processes-selecting relevant critical areas, strategy and planning, risk analysis, risk assessment, risk mitigation, assessing and monitoring program, and risk management review. To develop the dynamic Trusted Third Party auditing key role of public auditability in the cloud storage security API. The main objective of this research question is to understand information security threats relevant in cloud computing. Considering the security and efficiency in data sharing, we introduced the trusted third party (TTP). In this paper aims at identifying SLA based information security metrics relevant in cloud computing. In this research investigates different Objects relevant in cloud policy models and each of them can be used as a component in the proposed framework. A role represents a specific function within an organization and can be seen as a set of actions or responsibilities associated with this function.

Keywords— Secure Data collection, Cloud Storage Services, Secure distributed storage, Third Party Auditor, Insider Access, Service Level Agreement

I Introduction

Cloud services, which are deployed as self-contained components, are normally partial solutions that must be composed to provide a single virtualized service to Cloud consumers. This composition of services should be carried out in a dynamic and automated manner to promptly satisfy consumer requirements. Data storage correctness or some time more generally referred as data integrity verification is one of chief Cloud security problems [1]. Data can be altered by unauthorized entity without intimating to data owner.

Cloud Computing provides an optimal infrastructure to utilise and share both computational and data resources whilst allowing a pay-per-use model, useful to cost-effectively manage hardware investment or to maximise its utilisation. Cloud computing also offers transitory access to scalable amounts of computational resources, something that is particularly important due to the time and financial constraints of many user communities [2].

Data storage correctness schemes can be classified into two categories (a) Without Trusted Third Party (TTP) and (b) with TTP, based on who makes the verification. In case of TTP, an extra Third Party Auditor (TPA), some time in form of extra hardware or cryptographic coprocessor is used. This hardware scheme provides better performance due to dedicated hardware for the auditing process but has some drawbacks such as single TTP resulting into bottleneck in the system, mutually agreeing on a common TTP where there are thousands of users across the globe. Due to such kind of reasons, we prefer an approach where the functionalities of TPA is integrated in form of client application and the application can be downloaded by cloud user from cloud server [3]. This client application provides all the cryptographic functionalities to achieve the goals of integrity, authentication and confidentiality. As this is a software approach, the performance of the overall system may not be comparable to dedicated hardware kind of TTP alternatives. To improve performance, we emphasize offline execution of computationally costly cryptographic algorithms [4].

The auditing community is aware that current practices for auditing cloud environments are inadequate. As compliance grows in importance, enterprises implementing clouds need to satisfy their auditors' concerns; especially since creating an identity for an individual virtual machine and tracking that virtual machine from creation to deletion creates challenges for even the most mature virtualized environments[5][6].

The clouds have different architecture based on the services they provide. The data is stored on to centralized location called data centers having a large size of data storage. The data as well as processing is somewhere on servers. So, the clients have to trust the provider on the availability as well as data security. The SLA is the only legal agreement between the service provider and client [7]. The only means the provider can gain trust of client is through the SLA, so it has to be standardize.

A key part of a Service Level Agreement deals with monitoring and measuring service level performance. Essentially, every service must be capable of being measured and the results analysed and reported. The benchmarks, targets and metrics to be utilized must be specified in the agreement itself. The service performance level must be reviewed regularly by the two parties.

Data Protection

Data stored in the cloud typically resides in a shared environment collocated with data from other customers. Organizations moving sensitive and regulated data into the cloud, therefore, must account for the means by which access to the data is controlled and the data is kept secure.

Any progress must first occur in a particular domain in our work focuses on an important class of widely used applications that includes e-mail, personal financial management, social networks, and business tools such as word processors and spreadsheets. The following criteria define this class of applications:

Provide services to a large number of distinct end users, as opposed to bulk data processing or workflow management for a single entity;

Use a data model consisting mostly of sharable units, where all data objects have access control lists (ACLs) with one or more users; and Developers could run the applications on a separate computing platform that encompasses the physical infrastructure, job scheduling, user authentication, and the base software environment, rather than implementing the platform themselves.

Cloud Data Protection as a Service

Currently, users must rely primarily on legal agreements and implied economic and reputation harm as a proxy for application trustworthiness. As an alternative, a cloud platform could help achieve a robust technical solution by making it easy for developers to write maintainable applications that protect user data in the cloud, thereby providing the same economies of scale for security and privacy as for computation and storage; and enabling independent verification token of the secure data seeds with platform's operation and the runtime state of applications on it, so users can gain confidence that their data is being handled properly. With multiple providers and competition, users can regain control over their data. With a clear, universal application programming interface (API) to token for secure data seeds cloud services and the user's authorization, Cloud Data Protection as Service providers would be able to access and manipulate the data on another Cloud Data Protection as service provider. Such an API would also enable pervasive systems to run seamlessly between providers and allow interaction between users on different providers.

In a cloud setting, the unit of access control is typically a sharable piece of user data for example, a document in a collaborative editor. Ideally, the system offers some analogous confinement of that data, restricting its visibility only to authorized users and applications while allowing broad latitude for what operations are done on it [7]. This can make writing secure data seeds systems easier for programmers because confinement makes it more difficult for buggy code to leak data or for compromised code to grant unauthorized access to data.

II. RELATED WORK

Cloud infrastructure management networks are how cloud providers access the infrastructure and manage the different components within that infrastructure. Only authorized administrators should have access to this network because control of the management interfaces of the individual virtualization hosts allows for complete control of all of the virtual machines on that host. Root access on this interface is analogous to having the keys to a physical rack of servers within a data center. Administrator access to the central management console that manages all of the different virtualization hosts within the cloud is analogous to having the keys to the datacenter and every rack within that datacenter [9]. Therefore, protection of these interfaces is of paramount importance, and a customer should never need direct access to any of the systems within this network.

The reason for isolating this traffic is two-fold. First, both VMware VMotion and IP storage traffic need very fast data rates for optimal performance. Furthermore, traffic travels over the network in clear text and is susceptible to an attacker sniffing sensitive information off the network. By fully isolating this network, an attacker would need physical access to this network to have any chance of successfully compromising this data.

With so much remote execution, cloud computing requires robust credential management that enable secure logins to multiple cloud services from multiple devices in a seamless manner. The password schemes currently employed are a burden on users and have practically forced users into poor practices. Generally, users can remember a small number of passwords, yet each Web resource generally requires users to develop a unique set of credentials [10]. Services such as OpenID, which allow users to have a single set of credentials for multiple sites, are powerful, but may be inappropriate for sensitive institutions such as banks or government sites. Users may instead be able to use one-time-password devices, but they would need to have a unique device for each remote site to prevent one site from being able to use the credentials to authenticate to another.

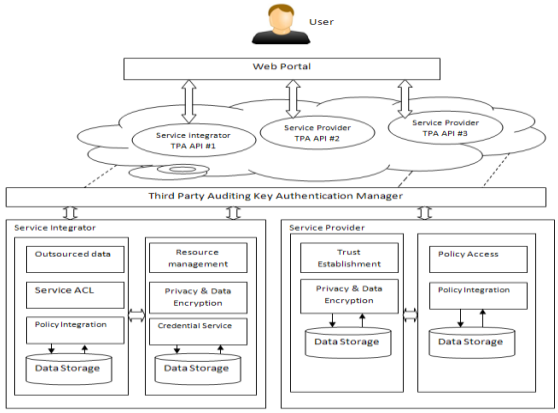
Even though these cloud computing components and characteristic provide compelling solutions to IT problems and many advantages, cloud computing is not risk-free or completely secure. Management is responsible for taking care of security risks to protect systems and data. Governance, risk and control of cloud computing are therefore critical in the performance of any assurance management process. Governance is enforced through the implementation of policies and procedures. These policies and procedures should be based on best practices and should be aligned between business and IT objectives. Risk identification and analysis is important to priorities the implementation (extent and time frame) of governance and controls, as well as to establish scope for reviewing or auditing cloud computing environments [11][12]. Based on the identification and analysis of risks, controls should be designed and implemented to ensure that necessary actions are taken to address risks and to achieve business and IT objectives [13]. This research aims to provide some guidelines to assist management with the identification of risks and recommendations for the mitigation of cloud computing security risks.
The author presented a P2P backup scheme in which blocks of a data file are dispersed across m+k peers using an (m+k,m)-erasure code. Peers can request random blocks from their backup peers and verify the integrity using separate keyed cryptographic hashes attached on each block. Their scheme can detect data loss from free riding peers, but does not ensure all data is unchanged.[13] proposed to verify data integrity using RSA-based hash to demonstrate uncreatable data possession in peer-to peer file sharing networks. However, their proposal requires exponentiation over the entire data file, which is clearly impractical for the server whenever the file is large. [14] Author proposed allowing a TPA to keep online storage honest by first encrypting the data then sending a number of precomputed symmetric-keyed hashes over the encrypted data to the auditor [15].

III. Implementation of Auditing and Data Scalability Framework

We implement the scalability framework we have defined for composite Web services. We proposed a new model–driven methodology for the security testing of cloud environments, to support batch auditing for TPA upon delegations from multi-user. We also proposed new scheme that enables the data owner to delegate tasks of Cloud storage data file encryption and trusted user secret key update to cloud servers without disclosing security data contents or user access privilege information. The trusted third party auditing process will bring in no new vulnerabilities towards user cloud storage data privacy. Our new method combined the secret key based dynamic secure authenticator with random masking to achieve the privacy-preserving public cloud data auditing system. We use a TPM-based trusted cloud storage SSL based third party auditing with secured key authentication

With the help of the loader, our trusted hypervisor finally takes over the machine. Users can use TPM-based attestation to verify the software stack running on the physical machine. The code is divided in three main parts: (1) data unit manager that stores the definition and information of the data units that can be accessed.

During our experiments we observed a significant number of read operations on individual clouds that could not be completed due to some error. The first thing that can be observed from the table is that the number of measurements taken from each location is not the same. Cloud storage concern the user does not have control over data until he has been gain access. To provide control over data in the cloud data-centric security is needed. Before accessing the data it should satisfy the policy rules already defined. So cloud should enforce this scheme by using cryptographic approaches.



In this Paper use the RSA algorithm as a basis to provide Cloud data-centric security for shared storage remote cloud data:

C Select two prime numbers

C Calculate n = p*q.

C Calculate f(n) = (p-1)(q-1)

C Select e such that e is relatively prime to f(n) and less than f(n).

C Determine d such that de congruent modulo 1 (mod f(n)) and d<f(n).

C Public key = {e, n}, Private key = {d, n}

C Cipher text c = message e mod n

C Plain text p = ciphertext d mod n

Implementation of dynamic packet filtering

Restriction of all inbound and outbound traffic to that information specified in the documented and maintained list of ports and services

Prevention of direct wireless access to the cloud infrastructure

Prevention of internal address direct access to external interfaces

Install perimeter firewalls between confidential and configuration data and external interfaces where supported by the cloud host.

Installation of personal firewall software, solutions on external devices, such as computers, mobile computers, mobile devices, and so on, that interface with the cloud environment where supported by your cloud host.

Implement IP masks to prevent internal systems from being presented and identified to external entities.

Implement a firewall to isolate confidential information, and ensure that all confidential information is stored behind the firewall. During retrieval of data, it is decrypted after checking the generated private key with existing private key

Before providing managed services, a security-as-a-service provider must overcome many concerns. Security services must be independent of any platform, adaptable to constantly changing environments, and supportive of a virtualized environment. To meet all these seemingly divergent requirements, security as a service must maintain compatibility with the service offerings provided in the virtualized environment.

Input: (i) call_for_proposals from CAs or other BAs

Output: (i) Instantiation of a Request_evaluator behavior

```
BlockReceive(call_for_proposals(Reqi))
if (not visitedFor(Reqi)) then
Prepare and Send Proposal
BlockReceive(reply, timeout)
if (reply = accept_proposal) then
Instantiate a Request_evaluator(Reqi) behavior
Else
Start over
else
Send refuse message
Start over
```

Users are then made members of roles, thereby acquiring the roles' authorizations. User access to resources is controlled by roles; each user is authorized to play certain roles and, based on his own role he can perform accesses to the resources and operate them correspondingly. As a role organizes a set of related authorizations together, it can simplify the authorization management.

Key Generation Algorithm

Choose a and b: two distinct prime numbers.
Compute m = a b, Where m is used as the modulus for public and private keys.
Compute Φ (m) = (a-1)(b-1), Where Φ is function.
Choose an integer E such that, $1 < E < \Phi(m)$ and common divisor of (E, Φ (m)) = 1.
Determine D = 1/E mod Φ (m).
All the above values of public key and private key must be kept secret.

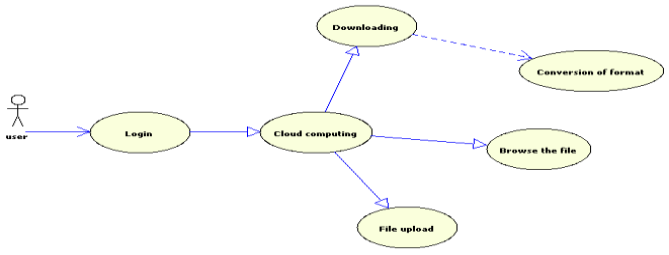
Encryption Algorithm

Sender A transmits her public key (m, E) to recipient B for the process of encryption data.
Represent the plaintext message as a positive integer n.
Computes the cipher c = nE (mod m).
Sends the cipher text c to recipient B

When the data owner redefines a certain set of attributes for the purpose of user revocation, he also generates corresponding proxy re-encryption keys and sends them to Cloud Servers. Cloud Servers, given these proxy re-encryption keys, can update user secret key components and re-encrypt data files accordingly without knowing the underlying plaintexts of data files. This enhancement releases the data owner from the possible huge computation overhead on user revocation. The data owner also does not need to always stay online since Cloud Servers will take over the burdensome task after having obtained the PRE keys.

Implementation Steps

The simulator is written in Java and runs on a Windows Vista Core 2 CPU 2.16 GHz machine. The set-up of parameters of message exchanges and network characteristics are as follows. The cloud deployment consists of 20 server nodes that are potentially able to host Web services. These 20 nodes are connected by a network such that the network has about 10%-50% network connectivity. In the simulation, we simulate the network of the cloud by randomly connecting one server node to the other in the network with a certain probability, which is equal to the network connectivity. For example, if the network connectivity is 20%, each node is directly connected to 30% of the other cloud nodes. We assume that there are 20 different types of request messages and response messages respectively exchanged between pairs of services during their interactions in composite services.



To ensure the security and dependability for cloud data storage under the aforementioned adversary model, we aim to design efficient mechanisms for dynamic data verification and operation and achieve the following goals: (1) Storage correctness: to ensure users that their data are indeed stored appropriately and kept intact all the time in the cloud. (2) Fast localization of data error: to effectively locate the malfunctioning server when data corruption has been detected. (3) Dynamic data support: to maintain the same level of storage correctness assurance even if users modify, delete or append their data files in the cloud. (4) Dependability: to enhance data availability against Byzantine failures, malicious data modification and server colluding attacks, i.e. minimizing the effect brought by data errors or server failures. (5) Lightweight: to enable users to perform storage correctness checks with minimum overhead.

We analyse the security strength of our schemes against server colluding attack and explain why blinding the parity blocks can help improve the security strength of our proposed scheme. With the appropriate runtime extraction the user-interface is able to migrate from the user's desktop to their mobile device and back again, without losing state. The ability to store data either locally or remotely in a transparent fashion will greatly help address issues raised in our previous work in personal data storage on the Internet. The control and implementation of policies is a business imperative that must be met before there is general adoption of cloud computing by the enterprise. SOA is derived from architecture and a methodology. Since cloud computing is typically driven from the view of business resources that are needed, there is a tendency to ignore the architecture. The second area that SOA brings to cloud computing is an end-to-end architectural approach.

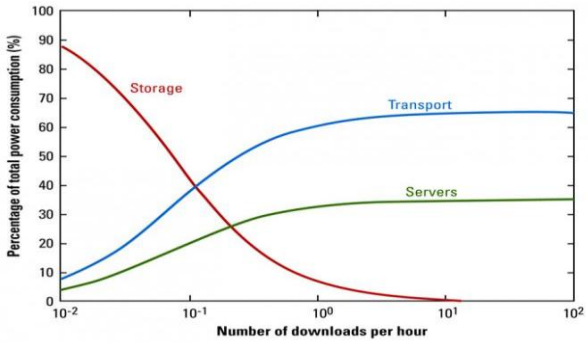
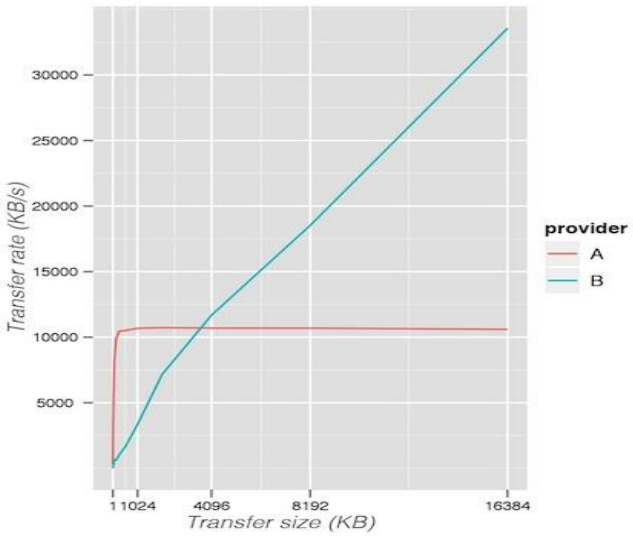


Figure shows that as the number of file transfers between the desktop and the cloud increases, the percentage of total power consumed in the transfer process increases. Says the report, "For a private cloud storage security service, at a download rates above one download per hour, servers consume 35%, storage consumes less than 7%, and the remaining 58% of total power is consumed in transport. These results suggest that transport dominates total power consumption at high usage levels for public and private cloud storage security services. The energy consumed in transporting data between users and the cloud is therefore an important consideration when designing an energy efficient cloud storage security service. Energy consumption in servers is also an important consideration at high usage levels. The percentage of total power consumed in servers is greater in private cloud computing than that in public cloud computing. In both public and private cloud storage security services, the energy consumption of storage hardware is a small percentage of total power consumption at medium and high usage levels. The proposed scheme is more suitable for the privacy-preserving of mass users.



The data is to be encrypted and compressed in multi-server. In encryption and compression the data that has to stored in a cloud can not be stored in a text format due to security reasons so it must be transformed into an encrypted format. The data also has to be compressed for secure transmission.

Conclusion

Finally, we investigated the problem of data security in cloud data storage, which is essentially a distributed storage system. To ensure the correctness of users' data in cloud data storage, we proposed an effective and flexible distributed scheme with explicit dynamic data support, including block update, delete, and append. We rely on erasure-correcting code in the file distribution preparation to provide redundancy parity vectors and guarantee the data dependability. By utilizing the homomorphic token with distributed verification of erasure coded data, our scheme achieves the integration of storage correctness insurance and data error localization, i.e., whenever data corruption has been detected during the storage correctness verification across the distributed servers, we can almost guarantee the simultaneous identification of the misbehaving server(s). Security design from the ground-up that promotes digitally signing each component-to-component call to allow the authorisation of all content executed by the user. When the data owner redefines a certain set of attributes for the purpose of user revocation, he also generates corresponding proxy re-encryption keys and sends them to Cloud Servers. Cloud Servers, given these proxy re-encryption keys, can update user secret key components and re-encrypt data files accordingly without knowing the underlying plaintexts of data files. When submitting their location information to the cloud, a blind user (and, in fact, any other user) could have security concerns that a malicious party could use this information to locate the user and harm or exploit the user for his own benefit.

REFERENCES

[1] C.Wang et al.,"Ensuring Data Storage Security in Cloud Computing," Proc. IWQoS '09, July 2009

[2] Q. Wang et al., "Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing," Proc. ESORICS '09, Sept. 2009, pp. 355–70

[3] C. Erway et al., "Dynamic Provable Data Possession," Proc. ACM CCS '09, Nov. 2009, pp. 213–22.

[4] C. Wang et al., "Privacy-Preserving Public Auditing for Storage Security in Cloud Computing," Proc. IEEE INFOCOM '10, Mar. 2010

[5] L. Carter and M. Wegman, "Universal Hash Functions," Journal of Computer and System Sciences, vol. 18, no. 2, pp. 143–154, 1979.

[6] J. Hendricks, G. Ganger, and M. Reiter, "Verifying Distributed Erasure coded Data," Proc. 26th ACM Symposium on Principles of Distributed Computing, pp. 139–146, 2007

[7] J. S. Plank and Y. Ding, "Note: Correction to the 1997 Tutorial on Reed-Solomon Coding," University of Tennessee, Tech. Rep. CS-03- 504, 2003.

[8] Q. Wang, K. Ren, W. Lou, and Y. Zhang, "Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance," Proc. of IEEE INFOCOM, 2009

[9] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple- Replica Provable Data Possession," Proc. of ICDCS '08, pp. 411–420, 2008.

[10] D. L. G. Filho and P. S. L. M. Barreto, "Demonstrating Data Possession and Uncheatable Data Transfer," Cryptology ePrint Archive, Report 2006/150, 2006, http://eprint.iacr.org/.

[11] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to Keep Online Storage Services Honest," Proc. 11th USENIX Workshop on Hot Topics in Operating Systems (HOTOS '07), pp. 1–6, 2007

[12] Dunlap, Kevin, and Rasmussen, Neil, "The Advantages of Row and Rack-Oriented Cooling Architectures for Data Centers", American Power Conversion, TCO

[13] Vouk, M.A. Cloud Computing - Issues, research and implementations, IEEE Information Technology Interfaces 30th International Conference, page(s): 31~40, June, 2008.

[14] Maithili Narasimha and Gene Tsudik. DSAC: integrity for outsourced databases with signature aggregation and chaining. Technical report, 2005

[15] C. Martel, G. Nuckolls, P. Devanbu, M. Gertz, A. Kwong, and S. Stubblebine. A general model for authenticated data structures. Technical report, 2001