Secure Data Transmission Using Cloud Computing

Trinath Naralasetty[#1], K . Eswar[*2]

[#1]PG Student ,St.Ann's College of Engineering and Technology,chirala.

*2Associate proessor,St.Ann's college of Engineering and Technology,Chirala.

Abstract— There is tension between user data protection and rich computation in the cloud. Users want to maintain control of their data, but also want to benefit from rich services provided by application developers using that data. At present, there is little platform-level support and standardization for verifiable data protection in the cloud. On the other hand, user data protection while enabling rich computation is challenging. It requires specialized expertise and a lot of resources to build, which may not be readily available to most application developers. We argue that it is highly valuable to build in data protection solutions at the platform layer

We propose a new cloud computing paradigm, *data protection as a service* (DPaaS) is a suite of security primitives offered by a cloud platform, which enforces data security and privacy and offers evidence of privacy to data owners, even in the presence of potentially compromised or malicious applications. Such as secure data using encryption, logging, key management.

***Keywords*— encryption, logging, key management, CSP, ACL.**

1. INTRODUCTION

Cloud computing promises lower costs, rapid scaling, easier maintenance, and services that are available anywhere, anytime. A key challenge in moving to the cloud is to ensure and build confidence that user data is handled securely in the cloud. A recent Microsoft survey found that "...58% of the public and 86% of business leaders are excited about the possibilities of cloud computing. But, more than 90% of them are worried about security, availability, and privacy of their data as it rests in the cloud."

Users want to maintain control of their data, but also want to benefit from rich services provided by application developers using that data. At present, there is little platform-level support and standardization for verifiable data protection in the cloud. On the other hand, user data protection while enabling rich computation is challenging. It requires specialized expertise and a lot of resources to build, which may not be readily available to most application developers. We argue that it is highly valuable to build in data protection solutions at the platform layer. The platform can be a great place to achieve economy of scale for security, by amortizing the cost of maintaining expertise and building sophisticated security solutions across different applications and their developers.

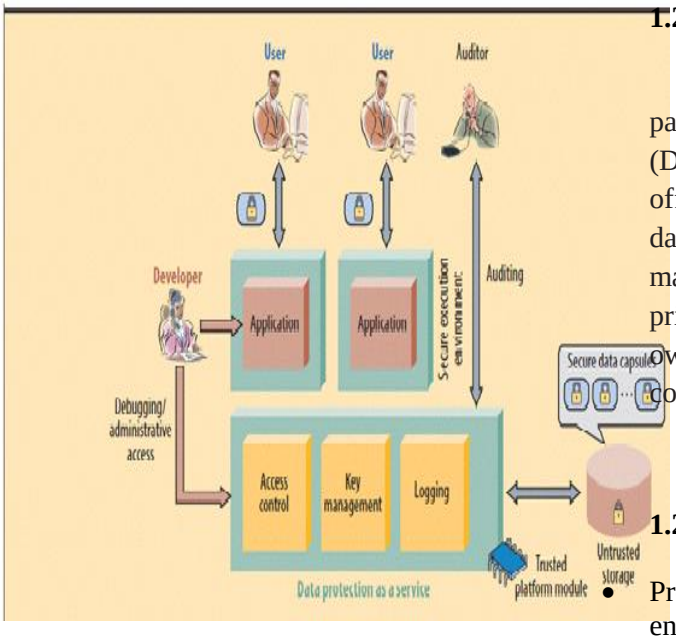


Fig: sample architecture for data protection

1.1EXISTING SYSTEM

In the existing system software developers who cannot afford store huge volumes of data on their local servers can upload their data to cloud. In the cloud data is stored across several servers. Each server is maintained by a specific cloud service provider (CSP). Software developers allow access to their data through their applications. Software developers write two sets of programs, one for providing access to data and another for checking whether data which is under the custody of CSP is secure or not.

1.1.1Draw backs:

1. Software developers have to build two sets of programs, they cannot concentrate on business logic.
2. Programs which check the integrity of data will be of same nature and in the existing system every software developer is forced to reinvent the wheel and develop same set of programs for their respective integrity checking mechanisms.

3. This introduces redundancy and wastes resource utilization for the whole IT industry

1.2 PROPOSED SYSTEM

We propose a new cloud computing paradigm, *data protection as a service* (DPaaS) is a suite of security primitives offered by a cloud platform, such as secure data using encryption, logging, key management, which enforces data security and privacy and offers evidence of privacy to data owners, even in the presence of potentially compromised or malicious applications.

1.2.1 Advantages:

- 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;
- 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.

2. MODULE DESCRIPTION

1. Cloud Storage Module

2. Trusted Platform Module

3. Third Party Auditor (Interface) Module

4. Data Owner (Developers' interface) Module

2.1Cloud Storage Module:

Cloud computing is the provision of dynamically scalable and often virtualized resources as a services over the internet Users need not have knowledge of, expertise in, or control over the technology infrastructure in the "cloud" that supports them. Cloud computing represents a major change in how we store information and run applications.

Instead of hosting apps and data on an individual desktop computer, everything is hosted in the "cloud"—an assemblage of computers and servers accessed via the Internet.

Cloud computing exhibits the following key characteristics:

1. Agility improves with users' ability to re-provision technological infrastructure resources.

2. Multi tenancy enables sharing of resources and costs across a large pool of users thus allowing for:

3. Utilization and efficiency improvements for systems that are often only 10–20% utilized.

4. Reliability is improved if multiple redundant sites are used, which makes well-designed cloud computing suitable for business continuity and disaster recovery.

5. Performance is monitored and consistent and loosely coupled architectures are constructed using web services as the system interface.

6. Security could improve due to centralization of data, increased security-focused resources, etc., but concerns can persist about loss of control over certain sensitive data, and the lack of security for stored kernels. Security is often as good as or better than other traditional systems, in part because providers are able to devote resources to solving security issues that many customers cannot afford. However, the complexity of security is greatly increased when data is distributed over a wider area or greater number of devices and in multi-tenant systems that are being shared by unrelated users. In addition, user access to security audit logs may be difficult or impossible. Private cloud installations are in part motivated by users' desire to retain control over the infrastructure and avoid losing control of information security.

Maintenance

Maintenance of cloud computing applications is easier, because they do not need to be installed on each user's computer and can be accessed from different places.

2.2 Trusted Platform Module:

Trusted Platform Module (TPM) is both the name of a published specification detailing a secure crypto processor that can store cryptographic keys that protect information, as well as the general name of implementations of that specification, often called the "TPM chip" or "TPM Security Device". The TPM specification is the work of the Trusted Computing Group.

Disk encryption is a technology which protects information by converting it into unreadable code that cannot be deciphered easily by unauthorized people. **Disk encryption** uses disk encryption software or hardware to encrypt every bit of data that goes on a disk or disk volume. Disk encryption prevents unauthorized access to data storage. The term "full disk encryption" (or **whole disk encryption**) is often used to signify that everything on a disk is encrypted, including the programs that can encrypt bootable operating system partitions. But they must still leave the master boot record (MBR), and thus part of the disk, unencrypted. There are, however, hardware-based full disk encryption systems that can truly encrypt the entire boot disk, including the MBR**.**

2.3 Third Party Auditor (Interface) Module

In this module, Auditor views the all user data and verifying data and also changed data.

Auditor directly views all user data without key. Admin provided the permission to Auditor. After auditing data, store to the cloud.

2.4 Data Owner (Developer's) Interface Module:

User store large amount of data to clouds and access data using secure key. Secure key provided admin after encrypting data. Encrypt the data using TPM. User store data after auditor, view and verifying data and also changed data. User again views data at that time admin provided the message to user only changes data.

3 Non Functional Requirements

3.1 External Interface Requirements:

User Interface: This tool provides an execellent user interface for us using which we can execute all the operations graphically.

Software Interfaces: These interface requirements should specify the interface with other. Software which the system will use or which will use the system, this includes the interface with the operating system and other applications.

The message content and format of each interface should be given.

Hardware Interfaces: Hardware interface is very important to the documentation. If the software is execute on existing hardware or on the pre-determined hardware, all the characteristics of the hardware, including memory restrictions, should be specified. In addition, the current use and load characteristics of the hardware should be given.

Performance Requirements

All the requirements relating to the performance characteristics of the system must be clearly specified. There are two types of performance requirements – static and dynamic. Static Requirements are those that do not impose constraint on the execution characteristics of the system. These include requirements like the number of terminals to be supported, and number simultaneous users to be supported, number of files, and their sizes that the system has to process. These are also called capacity of the system.

The processing speed, respective resource consumption throughput and efficiency measure performance. For achieving good performance Few requirements like reducing code, less use of controls, minimum involvement of repeated data etc., are to be followed. Each real-time system, software what provides required function but does not conform to performance of software requirements is acceptable. These requirements are used to test run time performance of software with the context of an integrated system.

3.2 Software Requirements

The minimum requirements to run the proposed system are

Operating System : Windows95/98/2000/XP

Application Server : Tomcat5.0/6.X

Front End : HTML, Java, Jsp

Scripts : JavaScript.

Server side Script : Java Server Pages.

Database : Mysql

Database Connectivity : JDBC.

3.3 Hardware Requirements

The minimum requirements to run the proposed system are

Processor - Pentium –III

Speed - 1.1 Ghz

RAM - 256 MB(min)

Hard Disk - 20 GB

Floppy Drive - 1.44 MB

Key Board - Standard Windows Keyboard

Mouse - Two or Three Button Mouse

Monitor - SVGA

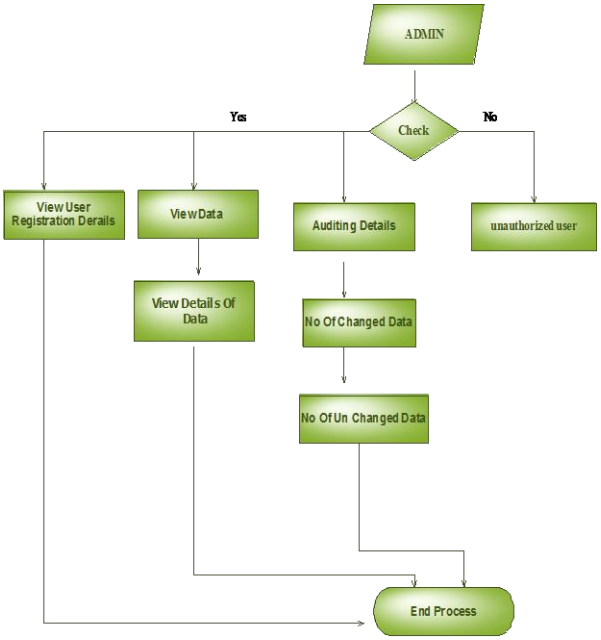
4. SYSTEM DESIGN

System design phase follows system analysis phase. Design is maintaining record proof design divisions and providing a blueprint for the implementation phase. Design is the bridge between system analysis and system implementation. System design is transition from a user oriented, document oriented to programmers or database personnel. The design is a solution, a "how to" approach to the creation a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Design goes through logical and physical stages of development, logical design reviews the present physical system, prepare input and output specifications, detail the implementation plan, and prepare a logical design walkthrough. System design is like a blue print for a building, it specifies all the features that are to be in the finished product. Design states how to accomplish objectives determined in the analysis phase.

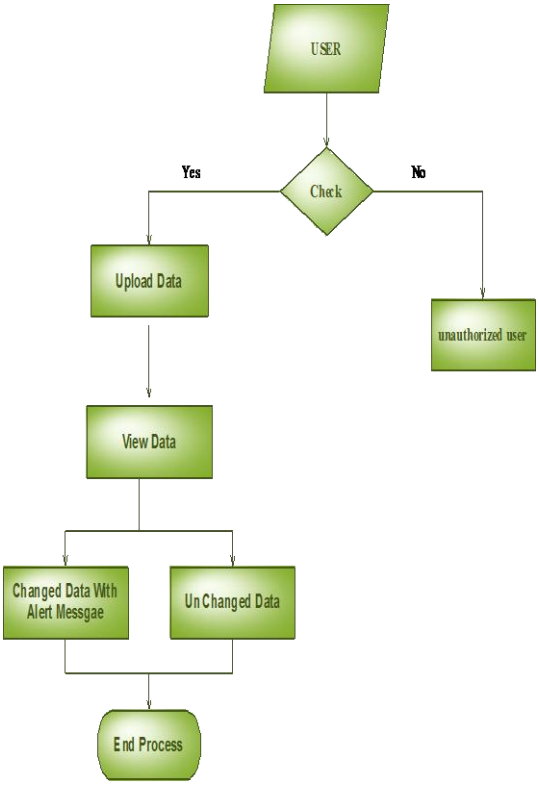
4.1 Data Flow Diagram

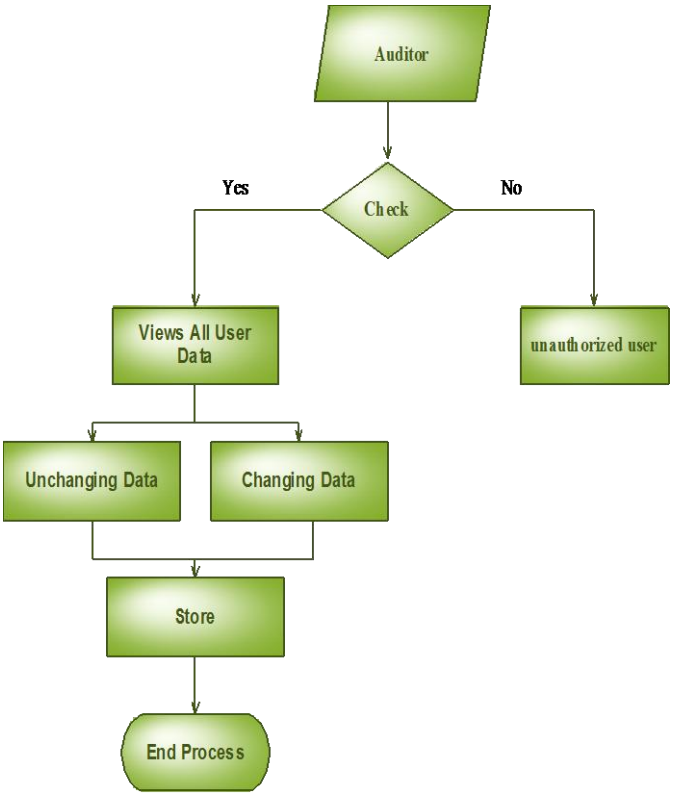
The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system.

4.2 System Design :(Admin)

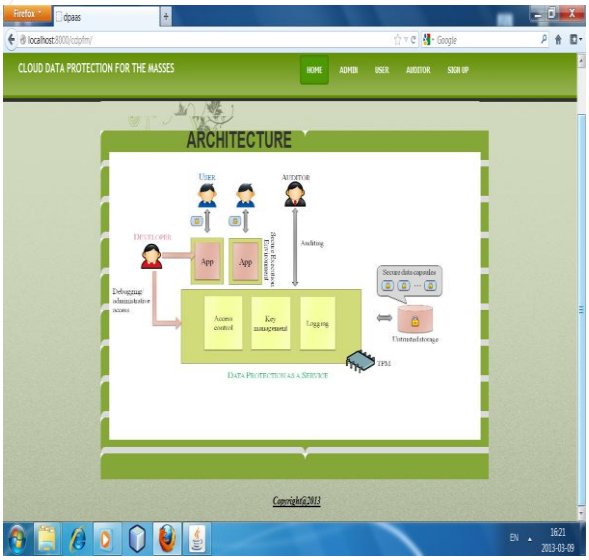


4.3User:





RESULT:



4.4 Auditor:

CONCLUSION

As private data moves online, the need to secure it properly becomes increasingly urgent. The good news is that the same forces concentrating data in enormous datacenters will also aid in using collective security expertise more effectively. Adding protections to a single cloud platform can

immediately benefit hundreds of thousands of applications and, by extension, hundreds of millions of users. While we have focused here on a particular, albeit popular and privacy-sensitive, classes of applications, many other applications also need solutions.

REFERENCES

[1] http://www.mydatacontrol.com.

[2] The need for speed. http://www.technologyreview.com/files/54902/GoogleSpeed charts.pdf.

[3] C. Dwork. The differential privacy frontier. In TCC, 2009.

[4] C. Gentry. Fully Homomorphic Encryption Using Ideal Lattices. In STOC, pages 169–178, 2009.

[5] A. Greenberg. IBM's Blindfolded Calculator. Forbes, June 2009. Appeared in the July 13, 2009 issue of Forbes magazine.

[6] P. Maniatis, D. Akhawe, K. Fall, E. Shi, S. McCamant, and D. Song. Do You Know Where Your Data Are? Secure Data Capsules for Deployable Data Protection. In HotOS, 2011.

[7] S. McCamant and M. D. Ernst. Quantitative information flow as network flow capacity. In PLDI, pages 193–205, 2008.

[8] M. S. Miller. Towards a Unified Approach to Access Control and Concurrency Control. PhD thesis, Johns Hopkins University, Baltimore, Maryland, USA, May 2006.

[9] A. Sabelfeld and A. C. Myers. Language-Based Information-Flow Security. IEEE Journal on Selected Areas in Communications, 21(1):5–19, 2003.

[10] L. Whitney. Microsoft Urges Laws to Boost Trust in the Cloud. http://news.cnet.com/ 8301-1009 3-10437844-83.html.