ENHANCED SECURITY MODEL BASED ON ID PATTERN IN PUBLIC CLOUD

PAVITHRAA.S[1] PHILOSHIYA MARSHELLINE ROSETTA.E[2] PUSHPA.S[3]

[1]Student, Computer Science and Engineering, New Prince Shri Bhavani Collage of Engineering and Technology, Tamilnadu, India.

[2]Student, Computer Science and Engineering, New Prince Shri Bhavani Collage of Engineering and Technology, Tamilnadu, India.

[3]Assistent Professor, Computer Science and Engineering, New Prince Shri Bhavani Collage of Engineering and Technology, Tamilnadu, Chennai.

ABSTRACT

Public Cloud Server (PCS) are likely to be used by more number of clients. New security issues must be handled in order to help more clients to process their data with no attempt of any information being stolen The restricted access to PCS makes the clients to delegate its proxy to process or upload their data and exchange them. On the other hand, remote data integrity checking is also a basic security issue out in the cloud. It makes the clients check whether their outsourced data is kept intact without downloading the whole data. From the security issues, a novel model for data exchanging and remote data integrity checking model in identity based open key cryptography: IDPUIC (character based middle person coordinating data exchanging and remote data integrity looking at in the open cloud). To strengthen the security model of ID-PUIC, clients are provided with an ID , for which the pattern of ID is only followed between Data owner and other user. Matching of ID if fails, a fake data is being generated to the unauthorized access. Fake data being generated is relevant to that of original data, by which the unauthorized clients have a chance to give up searching the original data. This reduces the chance of original data being read.

Keyword*: ID-PUIC-Identity-Based Proxy-Oriented Data Uploading and Remote Data Integrity Checking.*

1. INTRODUCTION

1.1 Introduction to Cloud Computing

Cloud computing allows movement of groups of servers that are remote and software networks that allows data storage in centralized manner and online access to computer services or resources, when they are needed. Clouds are classified as public, private or hybrid. The cloud is just a representation for the internet where different services such as server's storage and applications are delivered to computers in an organization and other devices through the internet. Cloud Computing is a type of computing that relies on sharing computing resources rather than having local servers or personal devices to handle applications. Cloud computing is otherwise called as 'on-demand computing', is a kind of computing in Internet, where resources that are shared, data and information are given to systems and other devices, when they are needed. It is a model for enabling ubiquitous, on-demand access to a shared pool of computing resources that are configurable. Cloud computing and solutions for storage gives users and companies with various capabilities to store and process their data in third-party data centers. It depends on sharing of resources to achieve consistency and careful management, similar to a usage over a network. At the foundation of cloud computing is the broader concept of converged infrastructure and shared services. Now a day's sharing data using cloud becomes normal in our life. If a user wants to share data, when they are in different place means they will go for cloud hosting. Cloud computing is a set of services. Cloud is mainly used for storage purpose and it can

be accessible anywhere at anytime. Because of this added advantage cloud computing technology is growing day by day. If a user hosted a data in a cloud means that particular data can be accessed by other user from any place.
The main enabling technology for cloud computing is virtualization. Virtualization software separates a physical computing devices into one or more "virtual " devices, each of which can be easily used and managed to perform computing tasks. With operating system-level virtualization essentially creates a expandable system of multiple individual computing devices, inactive computing resources that can be allocated and used more systematically. By minimizing user involvement, automation speeds up the process, reduces labor cost and reduces the possibilities of human errors. However, the complexity of security is greatly increased when data is distributed over a wider area or even a greater number of devices, as well as in multi tenant systems shared by unrelated users. In addition, user access to security audit logs may be difficult or impossible. Performance id monitored and consistent and loosely coupled architecture are constructed using web services as the system pool. Private cloud fitting are encouraged by users wish to posses control over the infrastructure and avoid losing control of security in the information.

1.2 INFRASTRUCTURE AS A SERVICE (IaaS)

In most fundamental cloud-service model and followed by the IETF(Internet Engineering Task Force) supplier of IaaS give computers physical or virtual machine and other resources. IaaS refers to online services that abstract the user from details of infrastructure like physical computing resources, location, data partitioning, scaling, security, backup etc. A hypervisor, such as Xen, Oracle Virtual Box, KVM, VMware ESX/EXSi, or Hyper-V runs the virtual machines as guests. Pools of hypervisors within the cloud operational system can support large number of virtual machines and the ability to scale services up and down according to customer varying requirements. IaaS clouds often offer additional resources such as virtual-machine disk-image library, raw block storage, file or object storage, firewalls, load balances, IP addresses, virtual local area networks (VLANs) and software bundles. IaaS-cloud suppliers provide these resources when needed from their large pools of equipments installed in data center. For wide-area connectivity, customers can use either the Internet or carrier clouds (dedicated virtual private networks).

1.3 PLATFORM AS A SERVICE (PaaS)

PaaS vendors offer a development environment to application developers. They provide typically developed toolkit and standard for development and channels for distribution and payment. In the PaaS models, cloud suppliers distribute a platform for computing, normally including operating system, programming language execution environment, database and web server. Application developers can thrive and run their solutions of software on the platform of cloud without the cost and difficulty of purchasing and managing the concealed hardware and software layers. With some PaaS offers like Azure in Microsoft and App Engine of Google, the concealed computer and storage resources expand automatically to match application demand so that the cloud user does not have to assign resources directly. The final has also been proposed by an architecture focusing to ease real-time in environments of cloud. Even more specific application types can be provided via PaaS, such as media encoding provided by services like bitcodin.com or media.io. Some integration and data management providers have also embraced specialized applications of PaaS as delivery models for data solutions. PaaS consumers do not manage or control the underlying cloud infrastructure including networks, servers, operating systems or storage, but have control over the applications that are moved and possibly configuration settings for the environment in which application is hosted.

1.4 SOFTWARE AS A SERVICE (SaaS)

In the model of software as a service (SaaS), users acquire access to application of software and databases. Cloud suppliers control the infrastructure and platforms that run the applications. SaaS is also referred to as "on-call software" and is usually rated on a pay-per-use basis or using a subscription fee. In the SaaS model, cloud providers install and operate application software in the cloud and the cloud users access the software from the cloud clients. Cloud users do not manage the cloud infrastructure and platform where applications runs. This removes the need to fix and run the application on the cloud user's personalized computers, that provides easy support and maintenance. Cloud applications differ from other applications in their expandability which can be achieved by copying tasks onto multiple virtual machines at run-time to meet work that can be changed on demand. Load balancers distribute the work over the set of virtual machines. This process is clear to the users in cloud, who sees only a single point of entry. To assist a large number of cloud users, cloud applications can be multitenant, meaning that any machine may serve more than one cloud-user organization.

2. PROBLEMS IN EXISTING SYSTEM

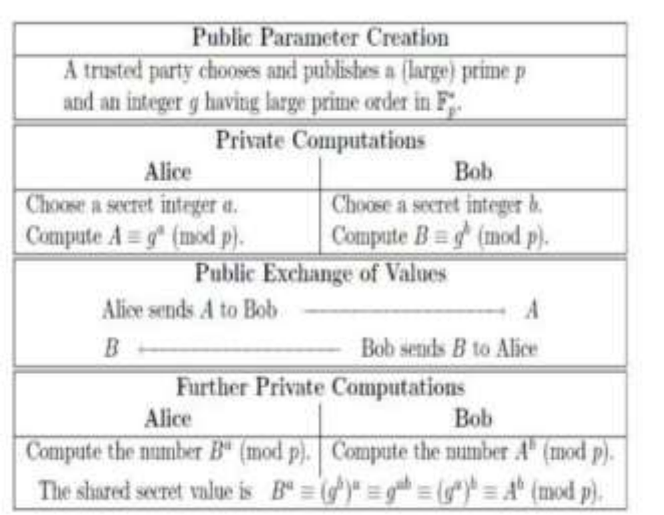
In Existing system , more clients would like to store their data to public cloud servers (PCSs) along with the rapid development of cloud computing. New security problems have to be solved in order to help more clients process their data in public cloud securely. When the client is restricted to access PCS, he will delegate its proxy to process his data and upload them . On the other hand, integrity checking on remote data is also an important security problem in public cloud storage. It makes the clients check whether their outsourced data are kept intact without downloading the whole data. From the security problems, a novel proxy-oriented data uploading and remote data integrity checking model in identity-based public key cryptography: ID-PUIC. We give the formal definition, system model, and security model. Then, a ID-PUIC protocol is designed using the bilinear pairings. The proposed ID-PUIC protocol is provably secure based on the hardness of computational Diffie–Hellman problem. The above ID-PUIC protocol though secure enough, in case of any unauthorized user randomly generates the original client's key, then the original data is being downloaded by that user.

3. PROPOSED SYSTEM

Security model for ID-PUIC in some cases may weaken, in case of any unauthorized access breaks through some security constraints. To avoid this, along with ID-PUIC a separate ID for clients are to be generated. By this, if any unauthorized access occurs the ID is verified to that of client's ID who request the file or data to be read. If the ID does not matches, then a duplicate file or data is then being generated to the unauthorized access.The duplicate file or data is relevant to that of original one. This can reduce the chance of original data being read.

3.1 Diffie-helman Key Exchange Algorithm

The idea of Diffie and Hellman is that it's easy to compute powers modulo a prime but hard to reverse the process.



Public Parameter Creation	
A trusted party chooses and publishes a (large) prime p and an integer g having large prime order in $\mathbb{F}_p^*$.	
Private Computations	
Alice	**Bob**
Choose a secret integer a. Compute $A \equiv g^a \pmod{p}$.	Choose a secret integer b. Compute $B \equiv g^b \pmod{p}$.
Public Exchange of Values	
Alice sends A to Bob $\longrightarrow$ A	
B $\longleftarrow$ Bob sends B to Alice	
Further Private Computations	
Alice	**Bob**
Compute the number $B^a \pmod{p}$.	Compute the number $A^b \pmod{p}$.
The shared secret value is $B^a \equiv (g^b)^a \equiv g^{ab} \equiv (g^a)^b \equiv A^b \pmod{p}$.	

Fig 1 Diffie hellman key exchange

Steps:

1. Two users, using insecure communication, agree on a huge prime p and a generator g.
2. User1 chooses some large random integer ie., secret key, $x_A < p$. Likewise User2 chooses secret key, $x_B < p$.
3. User1 computes its public key $y_A \equiv g^{x}{}_{A}$ (mod p) and sends it to User2 .User2 computes its public key
4. $y_B \equiv g^{x}{}_{B}$ and sends it to User1. Here $0 < y_A < p$, $0 < y_B < p$.
5. User1 computes $z_A \equiv y_B{}^{x}{}_{A}$ (mod p) and User2 computes $z_B \equiv y_A{}^{x}{}_{B}$ (mod p). Here $z_A < p$, $z_B < p$.

But $z_A = z_B$, since $z_A \equiv y_B{}^{x}{}_{A} \equiv (g^{x}{}_{B})^{x}{}_{A} = g^{(x_A\, x_B)}$ (mod p) and similarly,
$z_B \equiv (g^{x}{}_{A})^{x}{}_{B} = g^{(x_A\, x_B)}$ (mod p). So this value is their **shared secret key**.

3.2 SYSTEM ARCHITECTURE

In this paper , the important role for the user is to move login window to data owner window. This module has created for the security purpose. In the login page we have to enter login user id and password. It will check username and password is matching or not (valid user id and valid password). If we enter any invalid username or password we can't enter into login window , instead it will shows error message.So we are preventing from unauthorized user entering into the login window to user window. It will provide a good security for our project. So server containing user id and password server also check the authentication of the user. It will improves the security and preventing from unauthorized data owner enters into the network. And then ,it is used to help the user to upload the files. At the time of login, the user could be a valid user means only they are allowed to upload their files. Then, the Key Generation is used to help the Group member to encrypt the files and check their file is in safe by providing protection. Key Generation is the process for generating keys to our files. That key will have to be a unique for every group member while at the time of receives . If the file is in only view format ,then the Request is sended to the data owner for downloading the file. The data owner will check the request and if the user was an authorized person, the data owner response and provide key to the user The key is being generated to the data owner where the user gets the key when requested and the user can share and download the original file.If an unauthorized user generates a random key to access the file , thus occurrence of key mismatches leads to downloading of duplicate file.

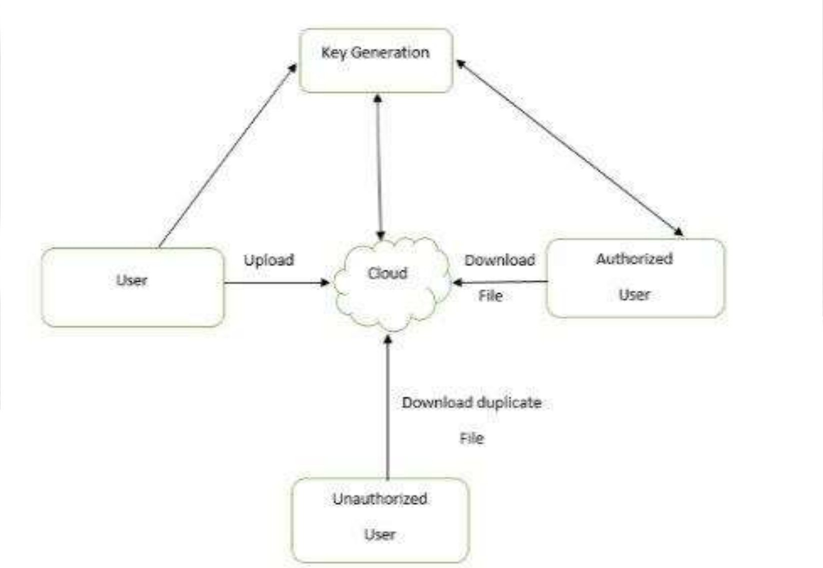


Fig 2 Architecture Diagram

4. LITERATURE SURVEY

Cloud storage is now a hot research topic in information technology. In cloud storage, data security properties such as data confidentiality, integrity and availability become more important. Nowadays, many Provable Data Possession(PDP) schemes are proposed to protect data integrity. In some cases, it has to delegate the remote data possession checking task to some proxy. However these PDP schemes are not secure since the proxy stores some information in cloud storage servers. Hence in this paper, we propose an efficient mutual verifiable provable data possession scheme, which utilizes Diffie-Hellman shared key to construct an homomorphic authenticator. In particular, the verifier in our scheme is stateless and independent of cloud storage service [1].

PRE schemes allows a proxy who has a re-encryption key to convert a cipher text originally encrypted for one party into a cipher text which can be decrypted by another party. The new security belief for PRE called Unforgeability of re-encryption key against collusion attack, UFReKey-CA for short. The proposed PRE schemes are claiming that their schemes meet UFReKey-CA. However , pointed out that the schemes do not meet UFReKey-CA . It is an open problem of constructing the scheme which meets UFReKey-CA. In this paper, we propose new PRE schemes which meet confidentiality assuming that the q-wDBDHI problem is hard and meet UFReKey-CA, assuming that 2-DHI problem is hard. [2]

In this paper, storage outsourcing is a rising trend which prompts a number of interesting security issues, many of which have been extensively investigated before. However , Provable Data Possession (PDP) is a topic that has only recently appeared in the research literature. The main issue is how to frequently and securely verify that a storage server is faithfully storing its client's outsourced data. The storage server is assumed to be unbelief, in terms of both security and reliability. (In other words, it might maliciously or accidently erase hosted data ; it might also relegate it to slow or offline storage) . [3]

Recently , cloud computing rapidly expanse as an alternative to conventional computing due to it can provide a flexible , dynamic and resilient infrastructure for both academic and business environments. In public cloud , the client moves its data to Public Cloud Server(PCS) and cannot control its remote data. Thus information security is an important problem in public cloud storage, such as data confidentiality, availability and integrity. In some cases, the client has no ability to check its remote data possession. It has to delegate the remote data possession checking task to some proxy. In this paper, we study proxy provable data possession (PPDP).[4]

Recent work on distributed, in-network aggregation assumes a benign general framework and threat model for the problem and present proof sketches, a compact verification mechanism that combines signatures of cryptography and Flajolet-Martin sketches to guarantee acceptable population of participants. Unfortunately, modern distributed system are plagued by malicious participants. In this paper we present towards verifiable yet efficient distributed, in-network aggregation in adversarial settings. We describe a aggregation error bounds with high probability. [5]

5. CONCLUSION

In this paper it proposes the novel security concept of ID-PUIC in public cloud. The paper strengthens the security model of ID-PUIC , clients are provided with an ID, for which the pattern of ID is only followed between Data owner and other user. Matching of ID pattern if fails, a duplicate data is being generated to the unauthorized access. Duplicate data being generated is relevant to that of original data, by which the unauthorized clients have a chance to give up searching the original data. This reduces the chance of original data being read.

6. REFERENCES

[1]Y. Ren, J. Shen, J. Wang, J. Han, and S. Lee, "Mutual verifiable provable data auditing in public cloud storage," *J. Internet echnol.*, vol. 16, no. 2,pp. 317–323, 2015.

[2]H. Guo, Z. Zhang, and J. Zhang, "Proxy re-encryption with unforgeable re-encryption keys," in

Cryptology and Network Security (Lecture Notes in Computer Science), vol. 8813. Berlin Germany: Springer-Verlag, 2014, pp. 20–33.

[3] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proc. SecureComm*, 2008, Art. ID 9.

[4]H. Wang, "Proxy provable data possession in public clouds," *IEEETrans. Services Comput.*, vol. 6, no. 4, pp. 551–559, Oct./Dec. 2013.

[5] J. Zhang, W. Tang, and J. Mao, "Efficient public verification proof of retrievability scheme in cloud," *Cluster Comput.*, vol. 17, no. 4, pp. 1401–1411, 2014.