Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 4, April 2014, pg.547 – 557

RESEARCH ARTICLE

An Aggregate Key Based Cryptosystem for Secure Data Sharing in Cloud Computing

R. Vanitha[1], V. Elavarasi[2]

[1]M.Tech Student, Department of Computer Science and Engineering, S.R.M. University Chennai

[2]Assistant Professor (OG), S.R.M. University Chennai

Abstract- Cloud computing provides the flexible architecture to share the applications as well as the other network resources. Cloud storage enables networked online storage when data is stored on multiple virtual servers generally hosted by third parties, rather than being hosted on dedicated servers. Key management and key sharing plays the main role in the data sharing concept of cloud computing. Traditional key cryptosystem lack the enhanced security techniques as the keys are generated by the exciting random key generation. Existing system said to have aggregate key cryptosystem in which key generated by means of various derivations of cipher text class properties of data and its associated keys. The aggregate was generated at only once, if we lost the key means it is difficult to access the data. So we introduce a SSH (Secure Shell) key, Digital signature, key escrow and encapsulation algorithm for secure authentication in cloud. This key is used to authenticate the remote computer and allow it to authenticate the user.

Index Keywords: Cloud computing, data sharing, aggregate key, SSH key, Key Escrow

1. INTRODUCTION

Cloud computing has become a significant technology trend either in the industrial or the academic field, and most of the experts expect that cloud computing will reshape -information technology (IT) processes 'and the IT market place. In Cloud Computing, users connect to the ’Cloud’, which appears as if it is a single entity as opposed to multiple servers. In this model, users can remotely store their data so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources Although this pay-per-use model of the cloud services brings significant savings for users and offers flexibility and scalability in terms of capacity and performance, it involves giving the cloud service provider (CSP) some form of control over the user's data.

In spite of the wide spread of cloud computing, different people evoke different perceptions about it. To some, it refers to accessing software and storing data in the "cloud" representation of the Internet or a network and using associated services. To others, it is seen as nothing new, but just a modernization of the time-sharing model that was widely employed in the 1960s before the advent of relatively lower-cost computing platforms. These developments eventually evolved to the client/server model and to the personal computer, which placed large amounts of computing power at people's desktops and spelled the demise of time-sharing systems. To formally describe cloud computing, the definition by the National Institute of Standards and Technology (NIST) is as follows:

"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." From the definition, we can conclude that the primary idea in cloud computing is that organizations no longer manage or own their data, but have it delivered as a service by a CSP. Over the last years, there is a trend to outsource more and more of data to external parties.

1.1 Cloud key characteristics:

a. On-Demand Self-Service: Cloud customer can make use of cloud resources without any human interaction between them and the cloud service provider (CSP).In addition; they can schedule, manage and deploy any of cloud services such as computation and storage when needed. This leads to reduction in the personnel overhead of the cloud provider, cut in costs of the offered services.

b. Broad Network Access: Cloud services are accessible over the network via standardized interfaces which enables users to access the services not only by complex devices such as personal computers, but also by light weight devices such as smart phones. In addition, the lowered cost of high-bandwidth network communication to the cloud provides access to a larger pool of IT resources that sustain a high level of utilization.

c. Location-Independent Resource Pooling: The cloud must be able to meet consumer's needs from resources. To do so, the cloud use a technique called "virtualization", which enables the cloud provider to pool his computing resources. This resource pool enables the sharing of virtual and physical resources by multiple consumers. As stated by NIST, "There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction.

d. Rapid Elasticity: It is the ability of the cloud to allocate and release resources quickly and efficiently in order to meet the requirements of the self-service characteristic of cloud computing. This automated process decreases the procurement time for new computing capabilities when the need is there, while preventing an abundance of unused computing power when the need has subsided.

e. Measured Service: Cloud computing can dynamically and automatically measure the used resources by cloud customers. These measurements can be used to bill the customer and provide them with a payment model based on "pay-per-use." The NIST view of measured service is "Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported providing transparency for both the provider and consumer of the utilized service."

1.2 Cloud service models:

One of the main principles of Cloud Computing is the `as-a-Service' paradigm in which some services are offered by a Cloud Service Provider (CSP) to customers for use. These offered services are often categorized using the SPI Service Model. This model represents the different layers/levels of service that can be offered to users by cloud service providers over the different application domains and types of cloud available. Clouds can be used to provide as-a-Service: software to use, a platform to develop on, or an infrastructure to utilize summarizes the SPI Service Model.

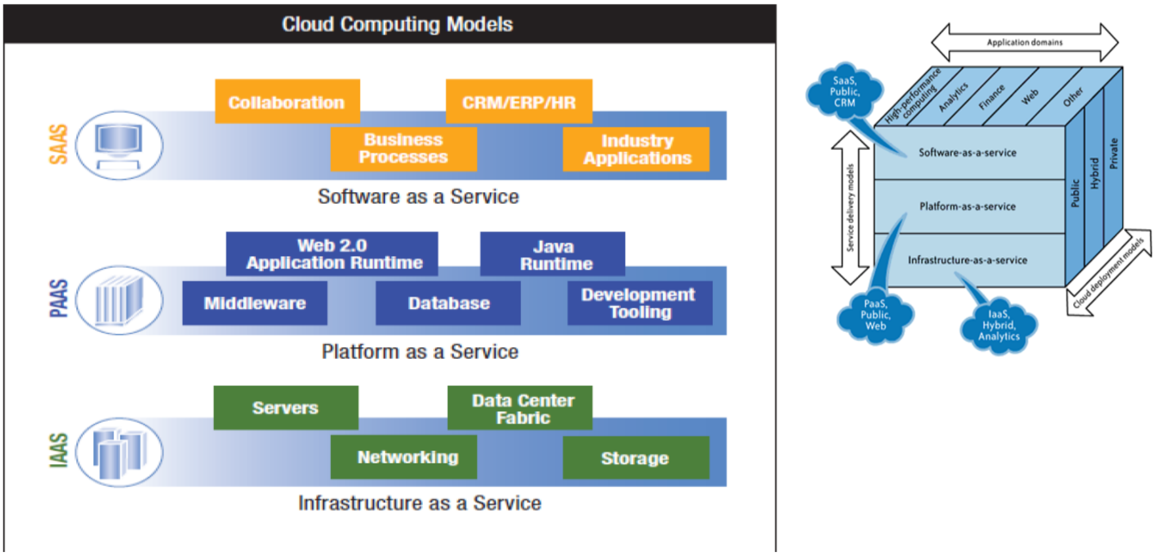


Fig:1.2 Cloud service models

1.2.1 Infrastructure as a Service (IaaS)

Infrastructure as a Service (IaaS) is a service that can provide the functionalities of a whole infrastructure including storage, networks, any platform and any number of desktops. The customers can make use of this service by configuring a virtual machine on the infrastructure, on which an operating system is installed. They deploy the middleware for communication with other applications, and install the CRM software. There is no need to buy extra servers, when the application needs more resources, extra CPUs and storage can be assigned via a web interface or via the CSP, the customers only pay for the used computing power and data storage.

1.2.2 Platform as a Service (PaaS)

In the Platform as a Service (PaaS) model, the CSP offers a development platform on top of the services delivered with IaaS. The CSP offers a development platform, on which applications can be built. In other words, software developers can develop their application through virtual development platform, accessible via a Web browser, without the need to install the software building tools on their own computer. This helps the developers to later distribute or deploy their apps to the cloud easily. In order to avoid confusion of this service with SaaS, it is good to imagine it as a cloud OS. The providers of the service enable its users to install their applications on a platform, which can provide any operating system or even emulate various types of hardware.

1.2.3 Software as a Service (SaaS)

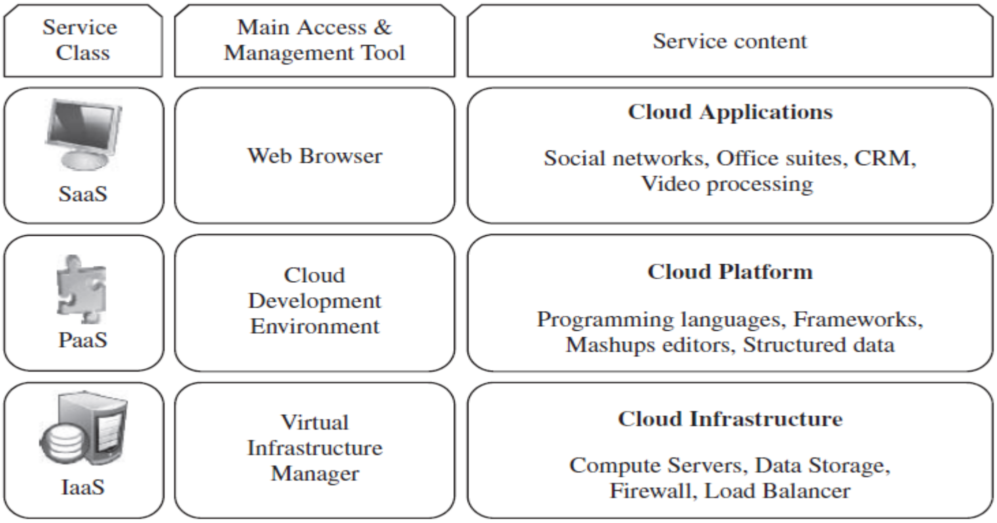
SaaS is a very popular service in which cloud service providers deliver software applications over the Web. A SaaS provider deploys their software, which is hosted on their own server infrastructure or use another vendor's hardware, on user's demand .This operation is usually done using a licensing model where applications may be licensed directly to an organization, group of users or, a user or, or through a third party that manages multiple licenses between user organizations, such as an ASP. The user then can be able to access the applications through any well defined and Internet device, which is most probably a Web browser.

Benefits of the SaaS Model:

1- It reduces the cost licensing, management hardware, and other resources required to internally host the application by outsourcing the application hosting to an independent software vendor (ISV).

2- It increase the control over the use of the software — by limiting the distribution of unlicensed copies and allowing the software vendor greater upgrade and patch management control.

3-It enables the provider to control and create multiple revenue streams with a one-to-many model leading to reduction in the duplication of software packages and overhead .It shows the three primary SPI framework services, paired with an example of the service the vendor supplies for that layer.



Service Class	Main Access & Management Tool	Service content
SaaS	Web Browser	**Cloud Applications** Social networks, Office suites, CRM, Video processing
PaaS	Cloud Development Environment	**Cloud Platform** Programming languages, Frameworks, Mashups editors, Structured data
IaaS	Virtual Infrastructure Manager	**Cloud Infrastructure** Compute Servers, Data Storage, Firewall, Load Balancer

Fig:1.2.3 Cloud Computing Stack

1.3 Cloud Deployment Models:

1. Private cloud

i) Infrastructure is operated solely for an organization

ii) It managed by the organization or by a third party

2. Community cloud

i) Supports a specific community

ii)Infrastructure is shared by several organizations

3. Public cloud

i) Infrastructure is made available to the general public

ii) Owned by an organization selling cloud services

4. Hybrid cloud

i) Infrastructure is a composition of two or more clouds deployment models

ii) Enables data and application portability

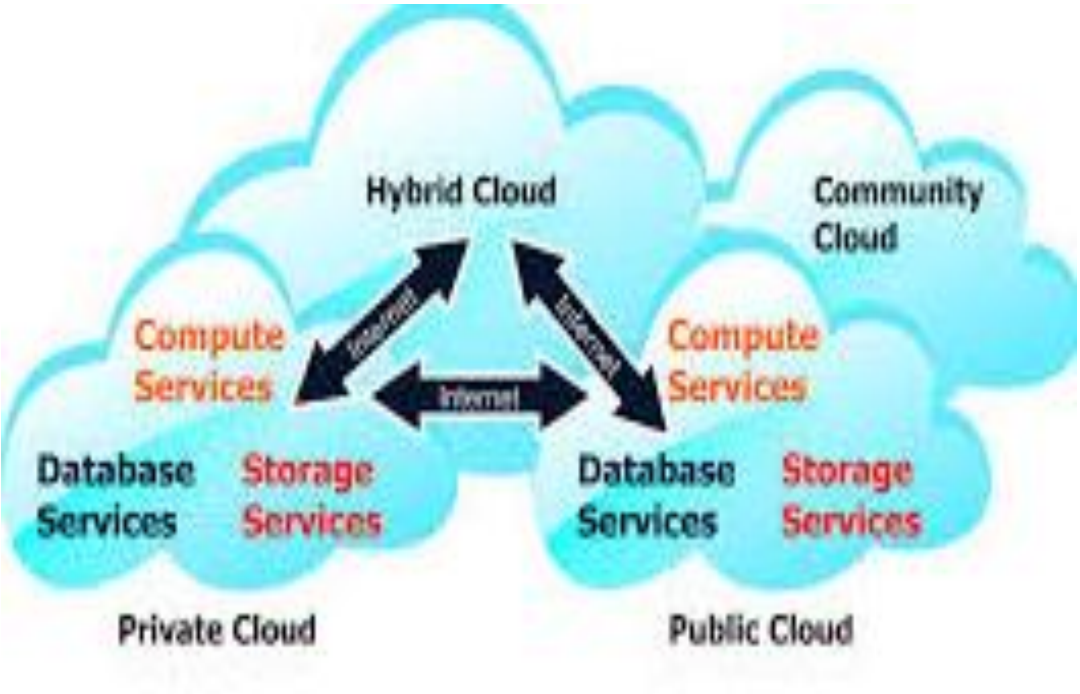


Fig: 1.3 Cloud models

Cloud Architecture:

The cloud is composed of a massive network of servers or even individual PCs interconnected in a grid. These computers operate in parallel, merging the resources of each computer to produce a power similar to that of supercomputers. In other words, the cloud is simply a collection of computers and servers that are publicly accessible via the Internet. These machines (computers and servers) can run any combination of operating systems; it's the processing power of the machines that matter. Although this architecture appears to be simple, it does require some intelligent management to connect all those computers together and assign task processing to multitudes of users. Below diagram shows the architecture behind a cloud computing system. As shown in the figure, it begins with user interface for the user to interact with the cloud and selecting a task or service (either starting an application or opening a document).After selecting the required service, a request is passed to the system management. In the system management, correct resources are found and then the appropriate provisioning services are called. Later on, these services choose the necessary resources in the cloud, launch the appropriate web application and either creates or opens the requested document. After that, the web application is launched and then the system's monitoring and metering functions track the usage of the cloud so that resources are apportioned and attributed to the proper user(s).

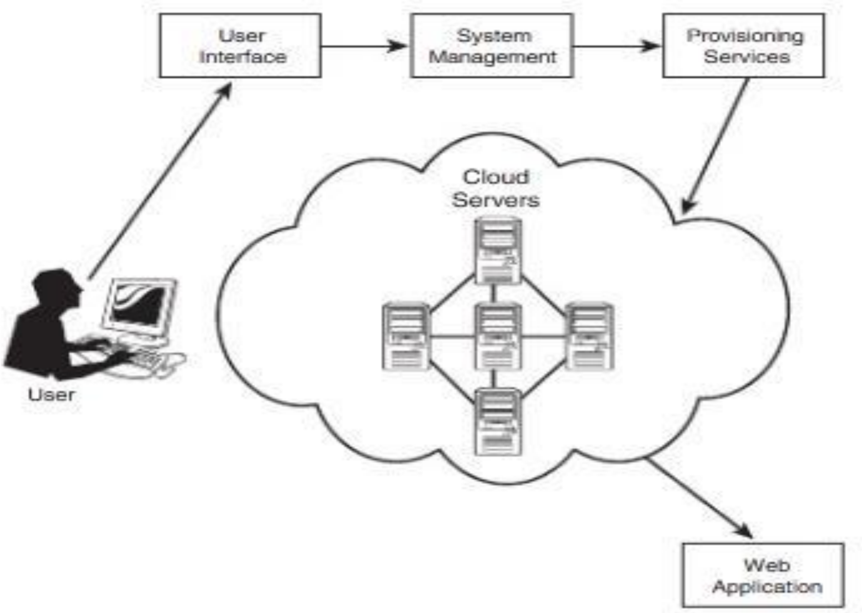


Fig: Cloud Architecture

2. RELATED WORKS

In [3], proposed aggregate signatures to compressing certificate chains. It given a certificate chain and some special additional Signatures. Aggregate signatures allow the compression of certificate chains without any additional signatures, but a verifier must still be aware of all intermediate links in the chain. We note that batch RSA also provides some signature Compression, but only for signatures produced by a single signer. Aggregate signature schemes give rise to simple verifiably encrypted signatures. These signatures enable user Alice to give Bob a signature on a message M encrypted using a third party's public key and Bob to verify that the encrypted signature is valid. Verifiably encrypted signatures are used in optimistic contract signing protocols to enable fair exchange.

In[10], proposed PRE schemes that are secure in arbitrary protocol settings, or in other words are secure against chosen ciphertext attacks. The concept of a CCA secure PRE scheme sounds almost self-contradictory, since on the one hand we want the cipher texts to be nonmalleable, and on the other hand we want to allow the proxy to “translate" the ciphertext from one public key to another. Still, we formulate a meaningful definition of CCA-secure PRE schemes, along with a construction that meets the definition in the standard model and under relatively mild hardness assumptions for bilinear groups.

In[4], proposed ABE schemes with constant-size cipher texts allowing for as expressive policies as possible. To this end, we propose several tradeoffs in terms of efficiency and expressivity. Our first result is to design a CP-ABE system for threshold policies with constant-size cipher texts and where the private key size is linear in the number of attributes held by the user. The scheme belongs to the cipher text-policy family in that the sender has the flexibility of choosing the threshold as he likes. The security is proved against selective adversaries under a non-interactive assumption. As a second contribution, we show that a certain class of identity-based broadcast encryption (IBBE) schemes readily yields KP-ABE schemes with monotonic access structures via a generic transformation. In a third step, we use a particular output of the aforementioned transformation to design a scheme supporting non-monotonic access structures without sacrificing the efficiency.

In[7], proposed the first identity-based broadcast encryption scheme with constant size cipher texts *and* private keys. Our construction is a Key Encapsulation Mechanism (KEM), thus long messages can be encrypted under a short symmetric key. In our solution, cipher texts and private keys are of constant size, and the public key is linear in the maximal value of *s*. Moreover, in our scheme, the Private Key Generator (*PKG*) can dynamically add new members without altering previously distributed information (as in IBE schemes). We also note that there is no hierarchy between identities, contrary to HIBE. The public key is linear in the maximal size of *S*, and not in the number of decryption keys that can be distributed, which is the number of possible identities.

In[12], proposed we use a simple scenario to introduce the challenging issues relating to group confidentiality and key management. We consider a source that sends data to a set of receivers in a multicast session. The security of the session is managed by two main functional entities: a Group Controller (GC) responsible for authentication, authorization and access control, and a Key Server (KS). To ensure confidentiality during the multicast session, the sender (source) shares a secret symmetric key with all valid group members, called Traffic Encryption Key (TEK). To multicast a secret message, the source encrypts the message with the TEK using a symmetric encryption algorithm.

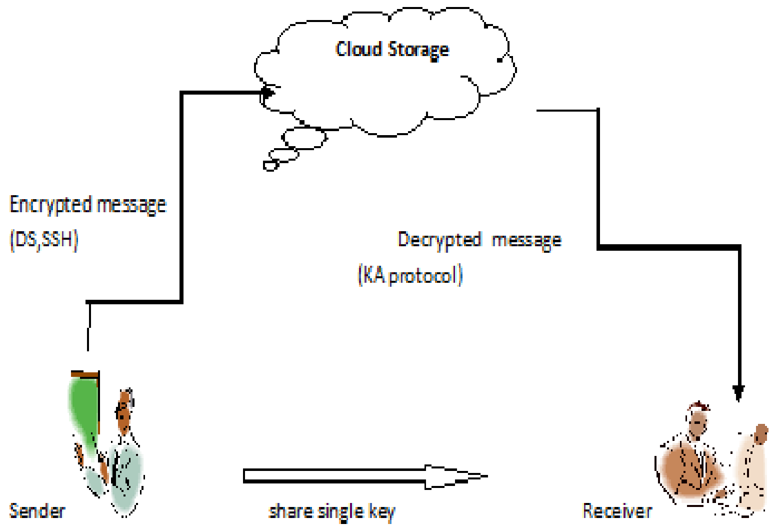
From the above papers, it is observed that how to share a secure data in cloud without lost the keys. In this paper , we introduce a novel Digital signature, SSH key, Hashing functions and key escrow algorithms. Compared with existing system we describe following features:

1. We store and share a secure data in cloud.
2. We use public key encryption, and create aggregate key for the data storing in cloud.
3. We add a digital signature to perform authentication in cloud.
4. The owner will perform the key escrow algorithm.
5. We propose a key Aggregate technique.

3. PROPOSED SCHEME

To solve the above problems, we propose key Escrow algorithm for prevent the keys. Our contributions are:

1. We propose Secure Shell key for additional security purpose. The owner will generate the key for encryption.
2. The decryption of multiple cipher text the size is constant in our scheme.
3. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, such that the sender cannot deny having sent the message and that the message was not altered in transit.
4. Key escrow systems provide a backup source for cryptographic keys



3.1Key Agreement Protocol:

Here we describe the framework and definition for key agreement protocol.

Framework:

In cryptography a key-agreement protocol is a protocol whereby two or more parties can agree on a key in such a way that both influence the outcome. If properly done, this precludes undesired third-parties from forcing a key choice on the agreeing parties.

The data owner generates the public, ssh,private key in key generation method. Based on keys we encrypt the message and store on cloud server. We decrypt the message using key agreement protocol.

3.1.1 System Parameters:

The SetUp process generates the system parameters. A user uses KeyGen to generate his public and secret key pair and ShareKeyGen to share his secret key to a set of m key servers.

3.1.2 Digital signature:

A valid digital signature gives a recipient reason to believe that the message was created by a known sender, such that the sender cannot deny having sent the message (authentication and non-repudiation) and that the message was not altered in transit.

3.1.3 Secure Shell:

SSH uses public-key cryptography to authenticate the remote computer and allow it to authenticate the user, if necessary. There are several ways to use SSH; one is to use automatically generated public-private key pairs to simply encrypt a network connection, and then use password authentication to log on.

3.1.4 Key Agreement protocol:

public-key agreement protocol that meets the criteria was the Diffie–Hellman key exchange, in which two parties jointly exponentiation a generator with random numbers, in such a way that an eavesdropper cannot feasibly determine what the resultant value used to produce a shared key is.

Exponential key exchange in and of itself does not specify any prior agreement or subsequent authentication between the participants. It has thus been described as an anonymous key agreement protocol.

3.1.5 Encipher:

The cryptographic transformation of data (plaintext) into a form (cipher text) that conceals the data's original meaning to prevent it from being known or used.

3.1.6 Decipher:

The cryptographic transformation of data (cipher text) that restores encrypted data to its original state (plaintext).

3.1.7 Key pair recovery:

There is sometimes a business case for recovery of private signing keys, for example, the user may forget his password and therefore be unable to access his private key. Where this is the case, there are two classes of key recovery techniques: key escrow and key encapsulation, with each technique having its own merits.

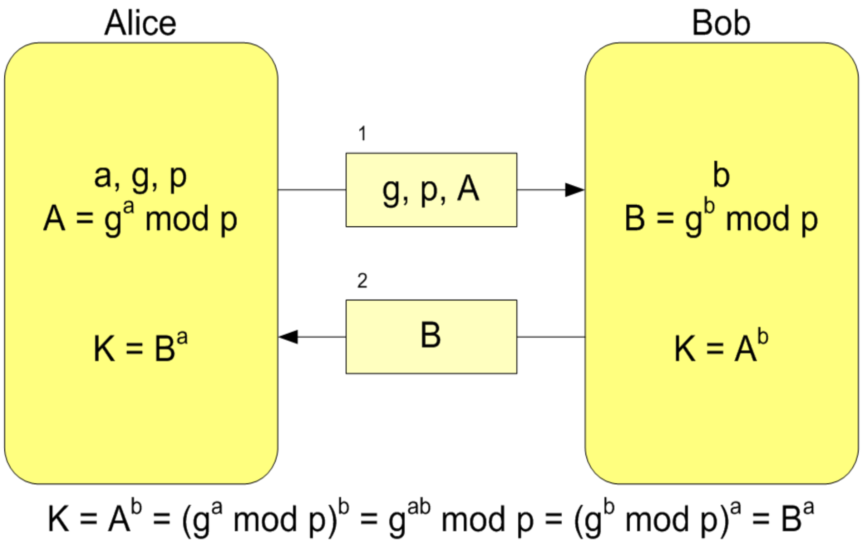
3.2 Public Key agreement Protocol:

It allows you to establish a key with a completely unknown individual and assumes each has a public key known to the other.

Diffie-Hellman: most famous key agreement protocol

•Discovered before RSA

•Original break-through in public-key cryptography.



We can use Diffie-Hellman with any algebraic group

–Z mod p

–Elliptic curve group

Security of DH

–Bounded by the "Computational Diffie-Hellman Problem"

•Given g, ga, gb, compute gab

–Related problem: "Decisional Diffie-Hellman Problem"

•Given g, ga, gb, z, determine if z = gab

–Currently DH is secure because we assume the DHP is difficult in the selected groups

•No known ways to solve DHP

•Best-known attack is bounded by DL problem.

3.3 Key Escrow Algorithm:

1. ***User Security Component (USC)***. This is a hardware device or software program that provides data encryption and decryption capabilities as well as support for the key escrow function. This support can include attaching a *data recovery field* (DRF) to encrypted data. The DRF may be part of the normal key distribution mechanism.

2. ***Key Escrow Component (KEC)***. This component, which is operated by *key escrow agents*, manages the storage and release or use of data recovery keys. It may be part of a public-key certificate management system or part of a general key management infrastructure.

3. ***Data Recovery Component (DRC).*** This consists of the algorithms, protocols, and equipment needed to obtain the plaintext from the ciphertext plus information in the DRF and provided by the KEC. It is active only as needed to perform a specific authorized data recovery.

3.4 Key Recovery:

Key recovery is based on hash functions. A cryptographic hash function is a mathematical transformation that takes an input message of arbitrary length and produces an output of fixed length, called the hash value. Hash functions guarantee good behavior of the hash function for any input pair; however, this refers to an average behavior over all keys and does not guarantee that each key yields a hash function with a uniform output distribution. For some schemes we identify rather large classes of weak keys that allow to easily forge authentication tags by swapping two blocks or by assigning specific values to some message blocks. The use of a weak key can typically be detected with a single text/MAC pair: it is sufficient to modify the text and submit a verification query. In principle the parties could check for the presence of weak keys, but in some cases this will substantially increase the complexity of the key generation procedure since a large number of combinations need to be avoided. Hash functions offer provable security, high speeds and parallelism; their simple combinatorial properties make them less robust than conventional message authentication primitives.

4. CONCLUSION

Our approach is more flexible and secure in cloud. A limitation in our work is so many keys are used in cloud. Escrow systems are somewhat risky because a third party is involved. SSH is important in cloud computing to solve connectivity problems, avoiding the security issues of exposing a cloud-based virtual machine directly on the Internet. An SSH tunnel can provide a secure path over the Internet, through a firewall to a virtual machine.

REFERENCES

[1] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy- Preserving Public Auditing for Secure Cloud Storage," *IEEE Trans. Computers*, vol. 62, no. 2, pp. 362–375, 2013.

[2] S.Kamara and K.Lauter,"Cryptographic Cloud Storage," Proc.Int'l Conf. Financial Cryptography and Data Security (FC), pp. 136-149, Jan. 2010

[3] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," in *Proceedings of Advances in Cryptology - EUROCRYPT '03*, ser. LNCS, vol. 2656. Springer, 2003, pp. 416–432.

[4] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)*. ACM, 2006, pp. 89–98.

[5] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing," Proc. IEEE INFOCOM, pp. 534-542, 2010.

[6] M. Chase and S. S. M. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption," in *ACM Conference on Computer and Communications Security*, 2009, pp. 121–130.

[7] F. Guo, Y. Mu, and Z. Chen, "Identity-Based Encryption: How to Decrypt Multiple Ciphertexts Using a Single Decryption Key," in *Proceedings of Pairing-Based Cryptography (Pairing '07)*, ser. LNCS, vol. 4575. Springer, 2007, pp. 392–406.

[8] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu,"Plutus: Scalable Secure File Sharing on Untrusted Storage," Proc. USENIX Conf. File and Storage Technologies, pp. 29-42, 2003.

[10] R. Canetti and S. Hohenberger, "Chosen-Ciphertext Secure Proxy Re-Encryption," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*. ACM, 2007, pp. 185–194.

[11] D. Boneh, R. Canetti, S. Halevi, and J. Katz, "Chosen-Ciphertext Security from Identity-Based Encryption," *SIAM Journal on Computing (SIAMCOMP)*, vol. 36, no. 5, pp. 1301–1328, 2007.

[12] Yacine Challal, Hamida Seba," Group Key Management Protocols: A Novel Taxonomy 2005 ISSN:1305-2403"

[13] Wang, C., Wang, Q., Ren, K., Lou, W.: Ensuring Data Storage Security in Cloud Computing. In: Proc. IEEE IWQoS. pp. 1–9 (2009)

[14] Zhu, Y.,Wang, H., Hu, Z., Ahn, G.J., Hu, H., Yau, S.S.: Dynamic Audit Services for Integrity Verification of Outsourced Storage in Clouds. In: Proc. ACM Symposium
On Applied Computing. pp. 1550–1557 (2011)

[15] Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: Proc. CRYPTO. pp. 41–55. Springer-Verlag (2004)