© 2015 IJSRSET | Volume 1 | Issue 2 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Data Sharing in Cloud Computing by Using Steganography

Dhanush Shri Vardhan. R, Immanuel Xavier.N, Illavarasan. S, Aravindan. B
Computer Science and Engineering, Dhanalakshmi College Of Engineering, Chennai, Tamilnadu, India

ABSTRACT

The important use of cloud computing is data sharing. However, security is always an issue in cloud. So, in this paper we show the ways to share our data securely, efficiently, and flexibly in cloud storage. We create new public-key cryptosystems that produce constant-size ciphertext such that efficient delegation of decryption rights for any set of ciphertext is possible. One can aggregate any set of secret keys and make them as compact as a single key, but encompassing the power of all the keys being aggregated. The secret key holder can release a constant-size aggregate key for flexible choices of ciphertext set in cloud storage, but the other encrypted files outside the set remain confidential. This compact aggregate key can be conveniently sent to others or be stored in a smart card with very limited secure storage. We provide formal security analysis of our schemes in the standard model. We also describe other application of our schemes. In particular, our schemes give the first public-key patient-controlled encryption for flexible hierarchy, which was yet to be known.

Keywords: patient-controlled encryption, data sharing, cloud storage, key-aggregate encryption

I. INTRODUCTION

Cloud computing provides data sharing and other resources through internet and gives access to many shared resources like networks, storage, servers. Service providers offers cloud platforms for their customer to use and create their web services, much like internet service providers offer customer high speed broadband to access the internet. There are numerous security issues for cloud computing as it encompasses many technologies. Security issues in cloud computing consists of application, platforms and infrastructure segment. Each segment performs different operations and offer different products for business and individuals around the world. The business application includes saas utility computing, web services, pass managed service providers, service commerce and internet integration. The cloud computing encounters various security issues, as it comprises of many technologies namely, networks, database, operating systems, virtualization, resources scheduling, transaction management, load balancing, concurrency control, memory management. Therefore security issues for many of these systems and technologies are applicable to cloud computing. For example -The network that interconnects the system in a cloud has to be secure -Mapping the virtual machines to physical machines has to be carried out securely -Data security involves encrypting the data as well as ensuring that appropriate policies are enforced for data sharing. Considering data privacy, a traditional way to ensure it is to rely on the server to enforce the access control after authentication (e.g., [3]), which means any unexpected privilege escalation will expose all data. In a shared-tenancy cloud computing environment, things become even worse. Data from different clients can be hosted on separate virtual machines (VMs) but reside on a single physical machine in a target VM could be stolen by instantiating another VM coresident with the target one. Regarding the availability of files, there are the series of cryptographic schemes which as go as far as allowing the third party auditor to check the availability of files on behalf on the data owner without leakage of anything about the data [4] or without compromising the data owner anonymity [5].A cryptographic solution ,for example [6], with proven security relied on number-theoretic assumption is more desirable, whenever the user is not perfectly happy with trusting the security of the VM or the honesty of the technical staff. These are motivated to encrypt their data with their own keys before uploading them to server. Users can download

the encrypted data from the storage, decrypt them, then send them to others for sharing, but it loses the value of cloud storage. Users should be able to delegate the access rights of the sharing data to others so that they can access these data from the server directly. Drop box is an example .Assume that data owner puts all her private photos on drop box and she does not like to share her photos to everyone.

Due to data leakage just relying on the privacy protection mechanisms provided by drop box, so she encrypts all the photos using her own keys before uploading .One day, Data owner friend i.e. data user asks her to share the photos taken over all these years which data user appeared in .Data owner can then use the share function of drop box, but the problem now is how to delegate the decryption rights for these photos to data user. A possible option data owner can choose is to securely send data user secret keys involved. In a proxy re-encryption scheme a semi-trusted proxy converts a cipher text for data owners into a cipher text for data requestor without seeing the underlying plaintext. The fundamental property of proxy re-encryption schemes is that the proxy is not fully trusted. Several efficient proxy re-encryption schemes that offer security improvements over earlier approaches, the primary advantage of our schemes is that they are unidirectional and do not require delegators to reveal their entire secret key to anyone – or even interact with the delegate – in order to allow a proxy to re- encrypt their cipher texts. In these schemes, only a limited amount of trust is placed in the proxy. Identity management is one of the most common services deployed within companies and organizations because of its key role in access control, authorization and accountability processes. However, it introduces an overhead in cost and time, and in most cases it requires specific applications and personnel for managing, integrating and maintaining this service the cloud offers an innovative opportunity of externalizing this workload; this is what has been called Identity Management as a service (IDaaS or IDMaaS).In a proxy re-encryption (PRE) scheme, a proxy is given special. Information that allows into translates a cipher text under one key into a cipher text of the same message under a different key. The proxy cannot, however, learn anything about the messages encrypted under either key. This paper propose a definition of security against chosen cipher text attacks for PRE schemes address the problem of obtaining PRE schemes that are secure in arbitrary protocol settings, or in other words are secure against chosen cipher text attacks[5].

II. METHODS AND MATERIAL

A. Safe Keeping In Cloud

Data security and privacy have always been the challenging fields and that is the reason these issues are upgraded periodically but frequently [1] and when cloud computing uniqueness is compared with previous computing approaches launched new security and privacy challenges and among which following are the most important.

i. Ensuring authorised access to user data even if it is a service provider
ii. Both cloud provider and its customer should share responsibility for privacy and security.
iii. Providing secure and efficient partitioning of virtualized and shared infrastructure (multi-tenant environment) among different customers.

According to The Cloud Security Alliance (CSA) [2] survey, they defined seven most important threats for cloud computing.

1. Abuse and Nefarious Use of Cloud Computing (IaaS, PaaS)
2. Insecure Interfaces and APIs (IaaS, PaaS, SaaS)
3. Malicious insiders (IaaS, PaaS, SaaS)
4. Shared Technology Issues (IaaS)
5. Data Loss or Leakage (IaaS, PaaS, SaaS)
6. Account or Service Hijacking (IaaS, PaaS, SaaS)
7. Unknown Risk Profile (IaaS, PaaS, SaaS)

The threats are equally important – and should reflect the critical threat concerns in Cloud Computing that organizations experience during their adoption processes. All of these above mentioned threats can be categorized in several ways, threats involving devices, threats involving softwares, threats involving techniques, threats involving hazards and in this article we have categorised these threats on platform and computing characteristics.

B. Steganography

Steganos - Covered, Graphein - Writing (Ancient Greek). Steganography is the art and science of hiding information by embedding messages within other, seemingly harmless messages. More commonly, steganography is used to supplement encryption. An encrypted file may still hide information using

steganography, so even if the encrypted file is deciphered, the hidden message is not seen. A. Steganography in Cloud Computing Steganography techniques can be used to provide a perfect tool for data exfiltration, to enable network attacks or hidden communication among secret parties. The aim of these techniques is to hide secret data (steganograms) in the innocent looking carrier e.g. in normal transmissions of users. In ideal situation hidden data exchange cannot be detected by third parties. Almost all digital file formats can be used for steganography, but the formats that are more suitable are those with a high degree of redundancy. Redundancy can be defined as the bits of an object that provide accuracy far greater than necessary for the object's use and display [Currie and Irvine]. The redundant bits of an object are those bits that can be altered without the alteration being detected easily [Anderson and Petit colas (1998)]. Image and audio files especially comply with this requirement, while research has also uncovered other file formats that can be used for information hiding. Hiding information in text is historically the most important method of steganography.

C. Merkle-Hash Tree

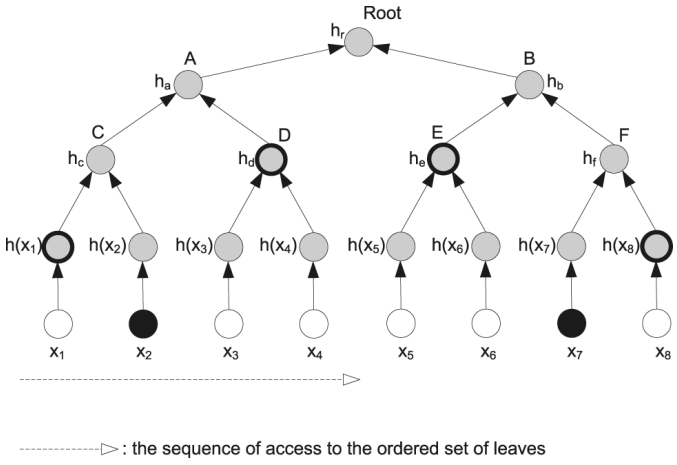

Figure 1: Merkle Hash Tree

Merkle hash tree is used here for additional security. To make One-Time Signature Schemes feasible, an efficient key management, that reduces the amount of public keys and their size, is needed. In [Mer79] Merkle introduced the Merkle Signature Scheme (MSS), in which one public key is used to sign many messages.

The Merkle Signature Scheme can only be used to sign a limited number of messages with one public key pub. The number of possible messages must be a 9 power of two, so that we denote the possible number of messages as N = 2n. The first step of generating the public key pub is to generate the public keys Xi and private keys Yi of 2n one-time signatures, as described in chapter 2. For each public key Yi, with $1 \leq i \leq 2n$, a hash value hi = H (Yi) is computed. With these hash values hi a Merkle Tree (also called hash tree) is build. We call a node of the tree ai, j, where i denote the level of the node. The level of a node is defined by the distance from the node to a leaf. Hence, a leaf of the tree has level i = 0 and the root has level i = n. We number all nodes of one level from the left to the right, so that ai, 0 is the leftmost node of level i. In the Merkle Tree the hash values hi are the leafs of a binary tree, so that hi = a0, i. Each inner node of the tree is the hash value of the concatenation of its two children. So a1, 0 = H (a0, 0||a0, 1) and a2, 0 = H (a1, 0||a1, 1). An example of a merkle tree is illustrated in figure1. In this way, a tree with 2n leafs and 2n+1 −1 nodes are build. The root of the tree and,0 is the public key pub of the Merkle Signature Scheme.

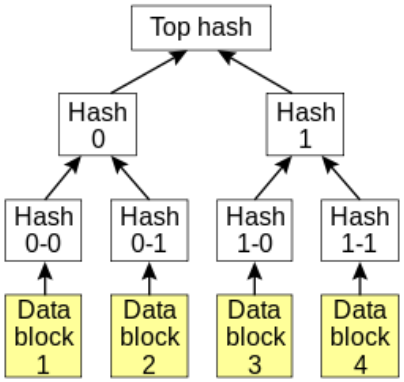

Figure 2: Merkle Tree

D. Creating Signature

To sign a message M with the Merkle Signature Scheme, the message M is signed with a one-time signature scheme, resulting in a signature sig′, first. This is done, by using one of the public and private key pairs (Xi,Yi,). The corresponding leaf of the hash tree to a one-time public key Yi is a0,i = H(Yi). We call the path in the hash tree from a0,i to the root A. The path A consists of n + 1 nodes, A0,...An, with A0 = a0,i being the leaf and An = an,0 = pub being the root of the tree. To compute this path A, we need every child of the nodes A1,...,An. We know that Ai is a child of Ai+1. To calculate the next node Ai+1 of the path A, we need to know both children of Ai+1. So we need the brother node of Ai. We call this node authi, so that Ai+1 = H(Ai||authi). Hence, n nodes auth0,...,authn−1 are needed, to compute every

node of the path A. We now calculate and save these nodes auth0,...,authn−1. How this is done efficiently is discussed in chapter 4. These nodes, plus the one-time signature sig′ of M is the signature sig = (sig′||auth2||auth3||...||authn−1) of the Merkle Signature Scheme Signature confirmation The receiver knows the public key pub, the message M, and the signature sig = (sig′||auth0||auth1||...||authn−1). At first, the receiver verifies the one-time signa- ture sig′ of the message M. If sig′ is a valid signature of M, the receiver com- putes A0 = H(Yi) by hashing the public key of the one-time signature. For j = 1,..,n−1, the nodes of Aj of the path A are computed with Aj = H(aj−1||bj−1).

III. RESULTS AND DISCUSSION

Merkle-Signature Method

If An equals the public key pub of the merkle signature scheme, the signature is valid.

Study Of Cost

Many signatures can be generated with using only one public key, this is the advantage of Merkle-signature scheme.. However, this advantage comes with an increase of computation time and signature length. In the following we will examine the computation time of each part of the signature process. To generate the public key pub, 2n one-time signature keys must be generated. Then every node of the hash tree must be computed. The tree consists of 2n+1−1 nodes. One hash operation is needed to calculate a node, so that 2n+1 −1 hash operations are needed to generate the public key. It is obvious, that the size of such a tree is limited. To compute 240 nodes is very costly, to compute 280 nodes is impossible. To generate a signature the nodes auth0,...,authn−1 are needed. If you do not store the nodes of the tree, the nodes must be generated again for every signature. Generating the tree is very expensive, so that generating the entire tree for every signature is impracticable for bigger trees. But saving all 2n+1−1 nodes would result in huge storage requirements. Hence, a good strategy is needed, to generate the signature without saving too many nodes, at a still efficient time. This problem is called the Merkle tree traversal problem. The verification time is quite fast, compared to the signature time. At first, the one-time signature must be verified. After that, the path A = A1,...,An must be computed. To do this, only n hash operations are needed, one for every node. The signature of the Merkle Signature Scheme consists of the one-time signa- ture sig′ and n nodes auth0,...,authn−1. If a 160 bit hash function is used, the signature size would be |sig| = |sig′|+ n∗160 bits.

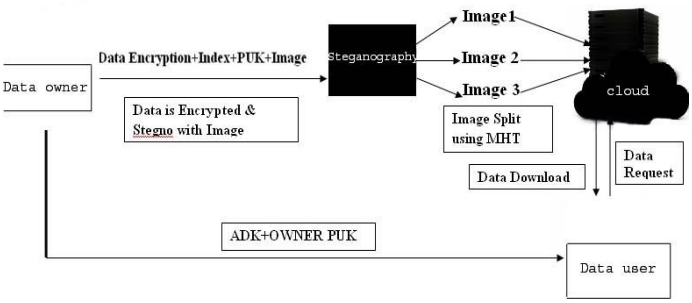

Figure 3: System Flow

IV. CONCLUSION

The problem with Cryptography is, Cryptographic schemes are getting more versatile and often involve multiple keys for a single application. Here, we consider how to "compress" secret keys in public-key cryptosystems which support delegation of secret keys for different cipher text classes in cloud storage. Our approach is more flexible than hierarchical key assignment which can only save spaces if all key-holders share a similar set of privileges. We are using steganography. Encrypted Outlet of original Data, Public Key and Index is made stegno into an Image. Data owner has to share the selected Image along with the ADK to download the Original Data. Remote Cloud would authenticate the Image along with the ADK to download Data which ensures security.

V. REFERENCES

[1] Beckham, J. (2011) The Top 5 Security Risks of Cloud Computing. Retrieved February17, 2012 from http://blogs.cisco.com/smallbusiness/the-top-5- security-risks-of-cloud- computing/

[2] Y. Chen, V. Paxson, R. H. Katz, What's New About Cloud Computing Security?, University of California, Berkeley, Technical Report No. UCB/EECS-2010-5, January 2010

[3] S.S.M. Chow, Y.J. He, L.C.K. Hui, and S.-M. Yiu, "SPICE – Simple Privacy-Preserving Identity-Management for Cloud Environment," Proc. 10th Int'l Conf. Applied Cryptography and Network Security (ACNS), vol. 7341, pp. 526-543, 2012.

[4] 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, Feb. 2013.

[5] R. Canetti and S. Hohenberger, "Chosen-Ciphertext Secure Proxy Re-Encryption," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 185-194, 2007.

[6] S.S.M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R.H. Deng, "Dynamic Secure Cloud Storage with Provenance," Cryptography and Security, pp. 442-464, Springer, 2012.