Impact Factor (SJIF): 3.632

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

Volume 3, Issue 7, July-2016

Privacy Preserving in Clouds By Using Decentralized Approach

Mrs. Roopa D. E.[1], Prof. B. N. Veerappa[2]

[1]M.Tech Student, Department of Studies in Computer Science and Engineering, UBDT,Davanagere,Karnataka,India

[2]Associate Professor, DOS in Computer Science and Engineering, UBDT, Davanagere,Karnataka,India

ABSTRACT: Cloud storage provides the on demand high quality hardware and software services so it will reduces the burden of hardware or software management locally .Even though cloud storage having so many benefits like on demand service ,pay as you use ,ease of access and maintainability ,encourages low investment etc one major issue that relinquishing user to go through cloud storage is security of remotely stored data specially in case of public cloud. To address this problem in this paper we are introducing a decentralized and distributed access of data .Here we are maintaining more than one Key distribution centre (KDC) to distribute secrete keys and attributes to the users. One KDC at a given time dedicated to a single user to avoid collusion .And to add even more security, we are taking the image file then converting that into plain text ,this plain text will be finally converted into cipher text using 16 bit Advanced encryption algorithm(AES) before giving to the cloud storage. In this scheme using Secure Hash algorithm for authentication purpose, SHA is the one of several cryptographic hash functions, most often used to verify that a file has been unaltered.

Keywords— Access control, Authentication, Secure hash algorithm, pailler algorithm.

I. I. INTRODUCTION

In this paper we propose a new decentralized access control scheme for secure data storage in clouds that supports anonymous authentication. The proposed scheme is resilient to replay attacks. A writer whose attributes and keys have been revoked cannot write back stale information. Distributed access control of data stored in cloud so that only authorized users with valid attributes can access them. Authentication of users who store and modify their data on the cloud.

The identity of the user is protected from the cloud during authentication. The architecture is decentralized, meaning that there can be several KDCs for key management. The access control and authentication are both collusion resistant, meaning that no two users can collude and access data or authenticate themselves, if they are individually not authorized. Revoked users cannot access data after they have been revoked. The proposed scheme is resilient to replay attacks. A writer whose attributes and keys have been revoked cannot write back stale information. The protocol supports multiple read and writes on the data stored in the cloud. The costs are comparable to the existing centralized approaches, and the expensive operations are mostly done by the cloud. Proposing privacy preserving authenticated access control scheme. According to our scheme a user can create a file and store it securely in the cloud. This scheme consists of use of the two protocols ABE and ABS.

The cloud verifies the authenticity of the user without knowing the user's identity before storing data. The scheme also has the added feature of access control in which only valid users are able to decrypt the stored information. The scheme prevents replay attacks and supports creation, modification, and reading data stored in the cloud**.**

Overview on Cloud Infrastructure

Figure 1 shows an overview on cloud infrastructure. Distributed computing is additionally known on-interest figuring, is a sort of web based registering that gives shared preparing assets and information to PCs and different gadgets on-interest. It is a model for empowering universal, on-interest access to shared pool of configurable processing assets. Distributed computing and capacity arrangements give clients and undertakings different abilities to store and process their information in outsider server farms. It depends on sharing of assets to accomplish lucidness and economies of scale, like a utility over a system.

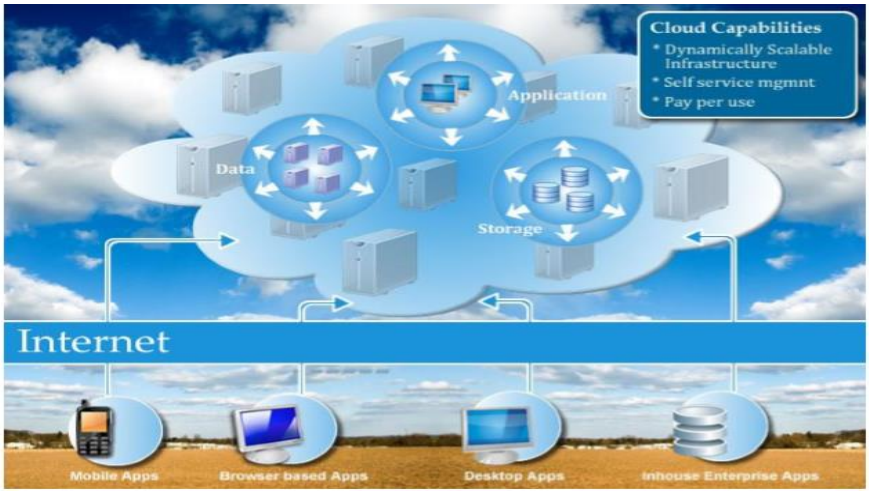

Fig 1 Overview of cloud computing

II. RELATED WORK

Secure outsourcing of computation to an untrusted (cloud) service provider is becoming more and more important. Pure cryptographic solutions based on fully homomorphic and verifiable encryption, recently proposed, are promising but suffer from very high latency. Other proposals perform the whole computation on tamper-proof hardware and usually suffer from the same problem. Trusted computing (TC) is another promising approach that uses trusted software and hardware components on computing platforms to provide useful mechanisms such as attestation allowing the data owner to verify the integrity of the cloud and its computation. However, on the one hand these solutions require trust in hardware (CPU, trusted computing modules) that are under the physical control of the cloud provider, and on the other hand they still have to face the challenge of run-time attestation.[1]

The core properties of a cryptographic storage service are that (1) control of the data is maintained by the customer and (2) the security properties are derived from cryptography, as opposed to legal mechanisms, physical security or access control. Therefore, such a service provides several compelling advantages over other storage services based on public cloud infrastructures.[2]

AES (Attribute-based Encryption) another set of cryptographic techniques that has emerged recently allows the specification of a decryption policy to be associated with a ciphertext. More precisely, in a (ciphertext-policy) attribute-based encryption scheme each user in the system is provided with a decryption key that has a set of attributes associated with it (this is how the "credentials" in Section 2 would be implemented). A user can then encrypt a message under a public key and a policy. Decryption will only work if the attributes associated with the decryption key match the policy used to encrypt the message.[2]

Cloud storage enables users to remotely store their data and enjoy the on-demand high quality cloud applications without the burden of local hardware and software management. Though the benefits are clear, such a service is also relinquishing users' physical possession of their outsourced data, which inevitably poses new security risks towards the correctness of the data in cloud. In order to address this new problem and further achieve a secure and dependable cloud storage service, we propose a flexible distributed storage integrity auditing mechanism, utilizing the homomorphic token and distributed coded data.[3]

To ensure anonymous user authentication Attribute Based Signatures were introduced by Maji*et al.* [4]. This was also a centralized approach. A recent scheme by the same authors [5] takes a decentralized approach and provides authentication without disclosing the identity of the users. However, as mentioned earlier in the previous section it is prone to replay attack.

III. PROPOSED SYSTEM

- Distributed access control of data stored in cloud so that only authorized users with valid attributes can access them.
- The identity of the user is protected from the cloud during authentication.
- The identity of the user is identified using Motherboard ID of a System.
- Motherboard ID + User Security key act as a key to generate a Cipher Text.
- The architecture is decentralized, meaning that there can be several KDCs for key management.
- The access control and authentication are both collusion resistant, meaning that no two users can collude and access data or authenticate themselves, if they are individually not authorized.
- The Data Integrity is computed by MAC authentication code.
- We are taking image file then converting image to plain text finally to cipher text .
- The protocol supports multiple read and writes on the data stored in the cloud.
- Authentication of users who store and modify their data on the cloud.
- After downloading file from cloud, user has to provide security key to decrypt a file contents and check integrity of a data.

IV. SYSTEM DESIGN

The architecture is decentralized, meaning that there can be several KDC's for key management. There are three users, a creator, a reader and writer. Creator Alice receives a token γ from the trustee, who is assumed to be honest. A trustee can be someone like the federal government who manages social insurance numbers etc. On presenting her id the trustee gives her a token γ.. For example, these can be servers in different parts of the world. A creator on presenting the token to one or more KDCs receives keys for encryption/decryption and signing. In the Fig. 2, SKs are secret keys given for decryption, Kx are keys for signing. The message MSG is encrypted under the access policy X. The access policy decides who can access the data stored in the cloud. The creator decides on a claim policy Y, to prove her authenticity and signs the message.

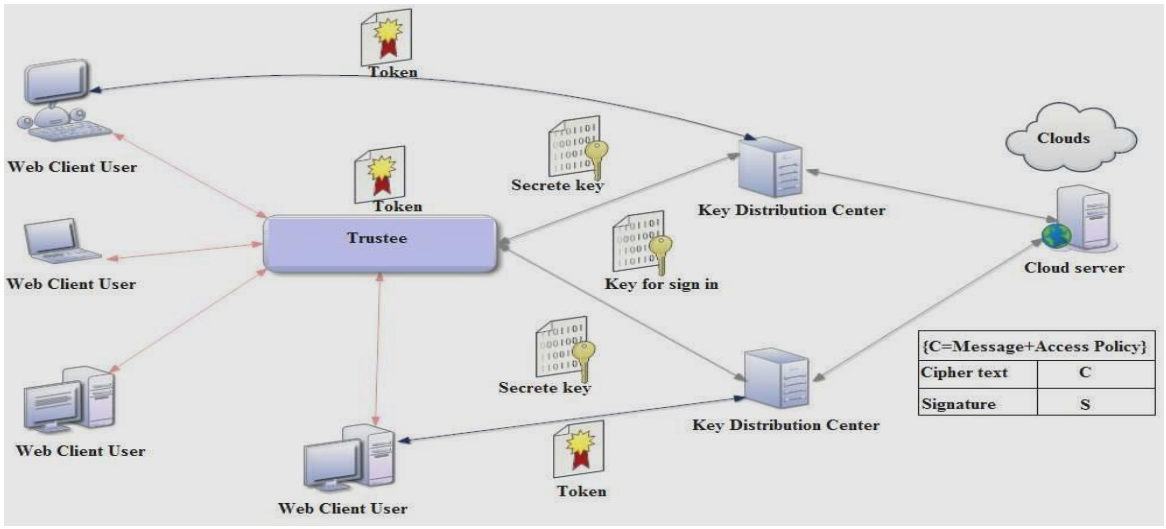

Fig 2.Architecture diagram

under this claim. The ciphertext C with signature is c, and is sent to the cloud. The cloud verifies the signature and stores the ciphertext C. When a reader wants to read, the cloud sends C. If the user has attributes matching with access policy, it can decrypt and get back original message. Write proceeds in the same way as file creation. By designating the verification process to the cloud, it relieves the individual users from time consuming verifications. When a reader wants to read some data stored in the cloud, it tries to decrypt it using the secret keys it receives from the KDCs. If it has enough attributes matching with the access policy, then it decrypts the information stored in the cloud.

A. *Creation of KDC :*

Different number of KDC's are created and to register a user details. KDC name, KDC id and KDC password are given as input to create KDC. Inputs will save in a database and to register a user details given a input as username and user id.

B. *KDC Authentication:*

After KDC given a user id to a user, the user will enrolled the personal details to KDC's given a input as user name, user id, password etc. The KDC will be verify the user details and it will insert it in a Database.

C. *Trustee and User Accessibility:*

Users can get the token from trustee for the file upload. After trustee was issuing a token, trustee can view the logs. User can login with their credentials and request the token from trustee for the file upload using the user id. After the user id received by the trustee, trustee will be create token using user id, key and user signature (SHA).

D. *Creation of access policy :*

After the key was received by the User, the message MSG is encrypted under the access policies. The access policies decide who can access the data stored in the cloud. The cipher text C with signature is c, and is sent to the cloud. The cloud verifies the signature and stores the cipher text C. When a reader wants to read, the cloud sends C. If the user has attributes matching with access policy, it can decrypt and get back original message and user can upload the file after user get key from the KDC.

E. *File accessing:*

Using their access policies the users can download their files by the help of kdc's to issue the private keys for the particular users. After trustee token issuance for the users, the users produce the token to the KDC then the token verify by the KDC if it is valid then KDC will provide the public and Private key to the user. After users received the keys the files are encrypt with the public keys and set their Access policies (privileges).

F. File Restoration :

Files stored in cloud can be corrupted. So for this issue, using the file recovery technique to recover the corrupted file successfully and to hide the access policy and the user attributes.

G. Secure Hash Algorithm

Definition: SHA-1 is one of several cryptographic hash functions, most often used to verify that a file has been unaltered. SHA is short for Secure Hash Algorithm. File verification using SHA-1 is accomplished by comparing the checksums created after running the algorithm on the two files you want to compare. SHA-1 is the second iteration of this cryptographic hash function, replacing the previous SHA-0. An SHA-2 cryptographic hash function is also available and SHA-3 is being developed.
One iteration within the SHA-1 compression function. A, B, C, D and E are 32bit words of the state. F is a nonlinear function that varies. n denotes a left bit rotation by n places. n varies for each operation. Wt is the expanded message word of round t. Kt is the round constant of round t. denotes addition modulo 232.

H. *Paillier Algorithm*

The Paillier cryptosystem, named after and invented by Pascal Paillier is a probabilistic asymmetric algorithm for public key cryptography. Key generation Choose two large prime number p and q randomly and independently of each other such that gcd (pq,(p-1)(q-1))=1. This property is assured if both primes are of equivalent length, i.e p,q {0,1 }s-1 for security parameter S . Compute n=pq and λ=lcm(p-1,q-1). Select random integer g where gZ*n2. Ensure n divides the

order of g by checking the existence of the following modular multiplicative inverse .μ= (L(gλ mod n2))-1mod n, where function L is defind as The public (encryption) key is (n,g).. The private (decryption) key is (λ,μ).

V. CONCLUSION

Our approach is reliable to distribute a key and provide a secure authentication and privacy for distributed data. We are using a MAC code to check Data Integrity. Decentralized approach is more feasible. Even any KDC is failure, we can generate a key using alternative KDC. And we are taking image file converting it into plain text then finally into cipher text before uploading to cloud ,this will adds another layer of security to user data. Hence our approach will encourages the use of public cloud services.

VI. ACKNOWLEDGEMENT

I am highly obliged to Department of computer science and engineering, UBDT college of engineering. And I am highly grateful and thankful to our guide Prof .B.N.Veerappa for his valuable instructions, guidance, corrections in my project work and presentation.

REFERENCES

[1] C. Wang, Q. Wang, K. Ren, N. Cao and W. Lou, "Toward Secure and Dependable Storage Services in Cloud Computing", IEEE T. Services Computing, vol. 5, no. 2, pp. 220–232, 2012

[2] S. Kamara and K. Lauter, "Cryptographic cloud storage," in Financial Cryptography Workshops, ser. Lecture Notes in Computer Science, vol. 6054.Springer, pp. 136–149, 2010.

[3] A.-R. Sadeghi, T. Schneider, and M. Winandy, "Token-based cloud computing," in TRUST, ser. Lecture Notes in Computer Science, vol. 6101. Springer, pp. 417–429, 2010

[4] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures: Achieving attribute-privacy and collusion-resistance," IACR Cryptology ePrint Archive, 2008.

[5] "Attribute-based signatures," in CT-RSA, ser. Lecture Notes in Computer Science, vol. 6558. Springer, pp. 376–392, 2011.

BIOGRAPHY

Prof. B. N. Veerappa Associate Professor in Department of Studies in Computer Science and Engineering, UBDT College, Davangere, Karnataka.

Roopa D. E. is a final year M.Tech student in Computer Science and Engineering, UBDT, Davangere, Karnataka. Her area of interests is in DBMS, web application.