RESEARCH ARTICLE OPEN ACCESS

Privacy Preserving using PAM in Cloud Computing

Ms. Shweta Dhavale[1], Ms. Pooja Mohan[2], Ms. Ashwini Shitole[3], Ms. Rohini Mote[4].

Department of Information Technology

Pimpri Chinchwad College of Engineering, Nigdi, Pune-411044.

Savitribai Phule Pune University, Maharashtra, India

Abstract:

The cloud user can remotely access software, services, application whenever they require over the internet. The user can put their data remotely to the cloud storage. So, It is necessary that the cloud must have to ensure data integrity and privacy of data of user.

The security is the major issue about cloud computing. The user may feel insecure for storing the data in cloud storage. To overcome this issue, here we are giving public auditing mechanism for cloud storage. For this, we studied Oruta system that providing public auditing mechanism. Revocation is all about the problems with security occur in system. And we are revoked these many problems from the system. We are also revoking existing members and adding new members in a group. In this way, we overcome the problem of static group. In this system, TPA is Third Party Auditor which maintains all the log credentials of user and it verifies the proof of data integrity and identity privacy of user. So, TPA plays a very important role in our system. Here we defining statement of our model as,**"Privacy Preserving using PAM in Cloud Computing ".**

.Keywords: Cloud Service Provider, Provable Data Possesion, Third Part Auditor, Public Auditing, Identity Privacy, Shared Data, Cloud Computing.

I. INTRODUCTION

The cloud service providers manage an enterprise class infrastructure that offers a secure, reliable and scalable environment for the users, at a very lower marginal cost due to the sharing nature of resources. It is very easy for the users to use cloud storage services to share data with others in a team, as data sharing becomes a standard feature in most cloud storage offerings a large storage space. It is possible if the data stored in an untrusted cloud can easily be lost or corrupted, it is only due to human errors and hardware failures. To protect the integrity or correctness of cloud data, it is best or easy to perform public auditing by introducing a third party auditor (TPA), who has the authentication to access and expose risk of cloud storage services on behalf of the users upon request.

The first [7] provable data possession [2] (PDP) mechanism to perform public auditing mechanism is designed to check the correctness of data stored in an untrusted server, without retrieving the entire data. We believe that sharing data among multiple users is perhaps one of the most engaging features that motivate cloud storage. A major unique problem introduced during process of public auditing for shared data in the cloud is how to preserve identity privacy from the TPA.

II. LITERATURE SURVEY

A. Existing system

The [7] provable data possession (PDP) mechanism to perform public auditing is designed to check the correctness that is integrity of data stored in an any untrusted server, without retrieving the entire data or file.

Moving a step forward, [8] Wang *et al.* (Referred to as WWRL) is designed to construct a public auditing mechanism for cloud data storage, so during public auditing, the content of private data or personnel data belonging to a personal user is not disclosed to the third party auditor.

[6][13]A. Juels and B.S. Kaliski,"PORs: Proof of Retrievability for Large Files". The public verifiability offered by [2] PDP/POR schemes can be naturally exploited to achieve POW. This phenomenon is called

"one stone, two birds". This scheme proposed notion of "Proof of Storage with Deduplication (POSD)".

In our model, we only consider how to audit the integrity of shared data in the cloud storage with static groups. It means the group is predefined before shared data is created in the cloud and the membership of users in the group is not changed during data sharing. The original cloud user is responsible for deciding that who is able to share her/his data before outsourcing data to the cloud. And then Another interesting problem is how to audit the correctness of shared data integrity of shared data in the cloud at dynamic groups, In dynamic group a new user can be added into the group and an existing group member can be revoked during data sharing while still preserving identity privacy.

B. Proposed system

We have only considered how to audit the integrity or correctness of shared data in the cloud with static groups. It means the group is already defined before shared data is created in the cloud and the membership of users that means adding and removing members in the group is not changed during data sharing.

We motivate the public auditing system of the data storage security in Cloud Computing and provide a privacy-preserving auditing task, i.e., our system supports an external auditor to audit user's outsourced data in the cloud without learning knowledge on the data content.

To the best of our knowledge, our scheme is the first to support scalable and efficient public auditing in the Cloud Computing. In general, our system achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA.

We improved the data security as well as the performance of our proposed schemes through concrete experiments and comparisons with the existing systems.

In our system, The user upload their documents then it is saved on the cloud server and the signature is save on the TPA simultaneously user download the document, verify it, block insertion, delete blocks that means the TPA mechanism that allows public auditing on shared data stored in the cloud. With this system, the identity of the signer on each block in shared data is kept private from a third party auditor (TPA), who is still able to verify the integrity of shared data without retrieving the entire file or data.

Our implementation results demonstrate the effectiveness and efficiency of our proposed mechanism when auditing shared data which is also shows the performance table that means the computation cost and RSA based instantiation.

C. Design Objectives

1. To construct the Web service system which would provide data integrity verification, provide encryption/decryption of the consumer data.

2. Here Defining access list for sharing data securely with specific band of individuals.

3. To construct thin client application which would call this service before uploading/downloading the data to and from the cloud.

III. ARCHITECTURE

The following figure shows the architecture of proposed system. In this architecture,

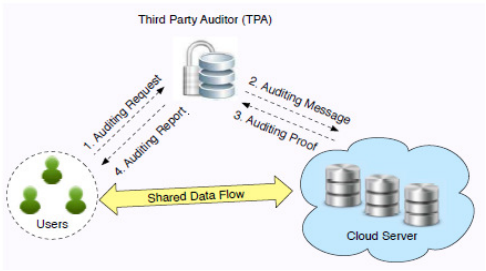


Fig. System Architecture

The followings are the system entities of our model,

- User : Store data in the cloud
- Third Party Auditor: TPA has the authentication to access and expose risk of cloud storage services on behalf of the users upon request.
- Cloud Service Provider : provide data storage service including storage space and computation resources.

In this above architecture, the user sends the auditing request to third party auditor when he wants to check the integrity of shared data. After receiving this request, TPA has generates the auditing message and send this message to CSP i.e., cloud service provider. And TPA retrieves the auditing proof of shared data from the cloud service provider. Then TPA verifies the correctness of the proof. If it is correct then TPA sends positive report to user otherwise he sends negative one.

IV. TECHNIQUES USED

A. Data Encryption Standard Algorithm

INPUT : plaintext m1 . . . m64; 64-bit key K=k1 . . . k64 (includes 8 parity bits).

OUTPUT : 64-bit ciphertext block C=c1 . . .c64.

1. (key schedule) Compute sixteen 48-bit round keys Ki, it is from K.

2. (L0, R0) = IP(m1, m2,. . .m64) (Use IP Table to permute bits; split the result into left and right 32-bit halves L0=m58,m50 . . . m8,R0=m57,m49 . . . m7)

3. (16 rounds) for i it is from 1 - 16, compute Li and Ri are as follows:

3.1. Li=Ri-1 3.2. Ri = Li-1 XOR *f* (R i-1, Ki) where *f*(Ri-1, Ki) = P(S(E(Ri - 1) XOR Ki)), computed as follows:

(a) Expand Ri-1 = r1,r2 . . . r32 from 32 to 48 bits, T =E(Ri-1).

(b) T ' =T XOR Ki. Represent T ' as eight 6-bit character strings: T '= (B1 . . . B8)

(c)T " =(S1(B1), S2(B2), . . . S8(B8)). Here Si(Bi) maps to the 4-bit entry in row r and column c of Si

(d)T"' =P(T"). (Use P per table to permute the 32 bits of T"=t1,t2 . . . t32, yielding t6,t7 . . . t25.)

4. b1,b2 . . . b64 = (R16, L16). (Exchange final blocks L16, R16.)

5. C = IP-1 (b1,b2 . . . b64).

6**.** End.

B] Digital Signature Algorithm (DSA)

The first part of the DSA algorithm is the public key and private key generation, which can be described as:

1) Choose a prime number q, which is called the prime divisor.
2) Choose another primer number p, such that p-1 mod q = 0. p is called the prime modulus.
3) Choose an integer g, such that 1 < g < p, g**q mod p = 1 and g = h**((p–1)/q) mod p. q is also called g's multiplicative order modulo p.
4) Choose an integer, such that 0 < x < q.
5) Compute y as g**x mod p.
6) Package the public key as {p,q,g,y}.
7) Package the private key as {p,q,g,x}.

The second part of the DSA algorithm is the signature generation and signature verification can be described as follow:

To generate a message signature, the sender follows these many steps:

1) Generate the message digest h, using a hash function algorithm likes SHA1.
2) Generate a random number k, such that 0 < k < q.
3) Compute r as (g**k mod p) mod q. If r = 0, select a different k.
4) Compute i, such that k*i mod q = 1. i is called the modular multiplicative inverse of k modulo q.
5) Compute s = i*(h+r*x) mod q. If s = 0, select a different k.
6) Package the digital signature as {r,s}.

To verify a message signature, the receiver of the message and the digital signature can follow these steps:

1) Generate the message digest h, using the same hash algorithm.
2) Compute w, such that s*w mod q = 1. w is called the modular multiplicative inverse of s modulo q.
3) Compute u1 = h*w mod q.
4) Compute u2 = r*w mod q.
5) Compute v = (((g**u1)*(y**u2)) mod p) mod q.
6) If v == r, the digital signature is valid.

C] RSA Algorithm

1) Generate two large random primes numbers , *p* and *q*, of approximately equal size such that their product n = pq is of the required bit length, for e.g. 1024 bits.
2) Compute n = pq and (phi) φ = (p-1)(q-1).
3) Choose an integer *e*, 1 < e < phi, such that gcd(e, phi) = 1.
4) Compute the secret exponent *d*, 1 < d < phi, such that ed ≡ 1 (mod phi).
5) The public key is (n, e) and the private key (d, p, q). Keep all the values d, p, q and phi secret. [We prefer sometimes to write the private key as

(n, d) because you need the value of n when using d. Other times we might write the key pair as ((N, e), d).]

6) n is known as the modulus.

7) e is known as the public exponent or encryption exponent or just the exponent.

8) d is known as the secret exponent or decryption exponent.

V. RELATED WORK

This scheme ensures the correctness of user's data in cloud was proposed by [8] C. Wang, Q. Wang, K. Ren, and W. Lou, C. Wang, Q. Wang, K. Ren. Their scheme achieves the storage correctness insurance and data or file error localization method, that is, when the data corruption has been detected during the storage correctness verification.

[6][13] A. Juels and B. S. Kaliski, "PORs- Proofs of Retrievability for data or Files". The public verifiability offered by [2]PDP/POR schemes can be naturally exploited to achieve POW. This phenomenon is called "one stone, two birds". This scheme proposes notion of Proof of Storage with Deduplication (POSD).

To evaluate the efficiency of Oruta in experiments[14]. To implement these complex cryptographic operations that we mentioned before, The GNU Multiple Precision Arithmetic (GMP) 2 library and Pairing Based Cryptography (PBC)3 library.

COMPARISION WITH EXISTING MECHANISM

Parameters	PDP	WWRL	Oruta	PAM
Identity privacy	Yes	Yes	Yes	Yes
Data Privacy	No	Yes	Yes	Yes
Public Auditing	No	No	Yes	Yes
Integrity	Yes	No	Yes	Yes
Confidentiality and Privacy	No	No	No	Yes

Table1: Comparison table

VI. PERFORMANCE

In this system, we here analyze the computation cost and communication costs of PAM, and then evaluate the performance of PAM in experiments.

A] Computation cost:

During an auditing, the public verifier that is cloud server first generates some random values to construct an auditing challenge, which only calculates a small cost in computation. Then, after receiving the public auditing challenge, the cloud server needs to compute an auditing proof then it will send this proof to TPA server.

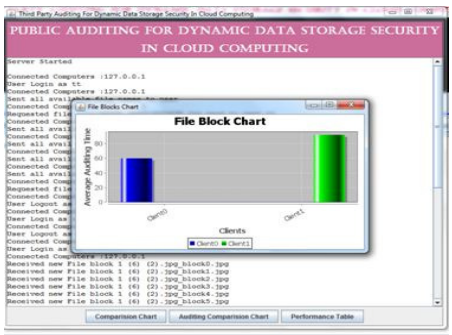
B] Communication cost:

The communication cost of PAM is mainly introduced by two aspects: the auditing challenge and auditing proof.

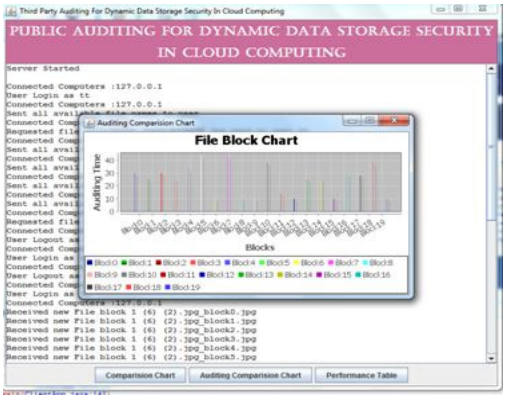
C] Experimental Result:

1. User can add in any group and remove whenever he wants. Dynamic group functionality achieved.

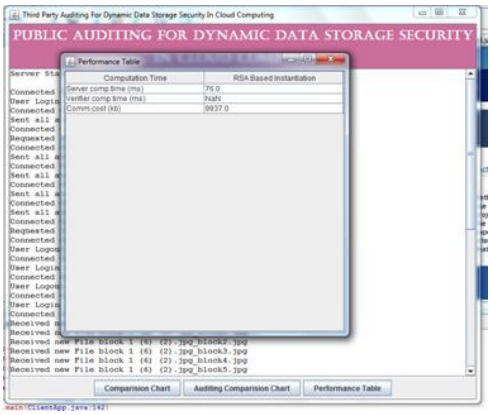
2. Performance of Auditing.



3. Performance of Batch Auditing



4. Performance table.



FUTURE SCOPE

1) All the logs credentials are maintained by TPA. So, it may possible that TPA misuse users log. So it is possible to revoke the TPA from this model.

2) The computation cost and communication cost will increase after revoking TPA. In future work, these problems will be solved.

CONCLUSION

Our Model ensures the Integrity of data sharing among users, and Identity of user is kept private from TPA in our model. TPA cannot retrieve the entire data of user. Revocation of user from the group and adding new user in the group is possible now after creating the group on cloud. It means the dynamic group is created.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D.Joseph, R. H.Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “A View of Cloud Computing”.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable Data Possession at Untrusted Stores”.

[3]C. Wang, Q. Wang, K. Ren, and W. Lou, “Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing”.

[4] R. L. Rivest, A. Shamir, and Y. Tauman, “How to Leak a Secret”.

[5] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, “Aggregate and Verifiably Encrypted Signatures from Bilinear Maps”.

[6] H. Shacham and B. Waters,“Compact Proofs of Retrievability”.

[7] Y. Zhu, H.Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S.Yau, “Dynamic Audit Services for Integrity Verification of Outsourced Storage in Clouds” .

[8] S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing” .

[9] D. Boneh, B. Lynn, and H. Shacham, “Short Signature from the Weil Pairing” .

[10] D. Boneh and D. M. Freeman, “Homomorphic Signatures for Polynomial Functions”.

[11] A. L. Ferrara, M. Green, S. Hohenberger, and M. Ø. Pedersen,“Practical Short Signature Batch Verification” .

[12] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data”.

[13] A. Juels and B. S. Kaliski, “PORs: Proofs pf Retrievability for Large Files”.

[14] Boyang Wang, Baochun Li and Hui Li, “Oruta :privacy preserving public auditing in cloud computing”