International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 17 (2015) pp 37531-37533

Ensuring Efficient Data Protection on Cloud by Distribution Concurrency

Uvaneshwari.M,
Department of IT, VelTech University, Avadi, Chennai-62, Tamil Nadu, India
uvaneshwarim@veltechuniv.edu.in.

Deepan. S,
Department of IT, VelTech University, Avadi, Chennai-62, Tamil Nadu, India.
deepan@veltechuniv.edu.in.

Ganesan. R,
Department of IT, VelTech University, Avadi, Chennai-62, Tamil Nadu, India.
ganeshitlect@gmail.com.

Naveen Raju. D,
Department of IT, VelTech University, Avadi, Chennai-62, Tamil Nadu, India.
naveenraju@veltechuniv.edu.in

Abstract
Cloud computing has been envisioned as the next-generation process. In according to that traditional solutions, where the IT services are under the personnel controls, cloud computing is moving the software and databases to the large data storage. The management regarding the services may not be fully trustworthy. The attribute, however, poses many security challenges which have not been well understood. The cloud data is an important aspect in quality of service. Unlike most prior works, the new scheme further supports secure and efficient dynamic operations that include data steps in it. The security analysis shows that the proposed scheme is highly efficient and resilient against malicious data modification attack & even server colluding attacks.

Introduction
Placing critical data in the hands of a cloud provider should come with the guarantee of security and available data in motion for use. Several alternate options exist for storage purpose, while data confidentiality solution service paradigm is still immature. We proposed an architecture that integrates cloud database services with data confidentiality and the possibility of executing concurrent operations on the data. This is the solution that supports geographically distributed clients to connect directly to an encrypted database, and to execute concurrently and the operations including the modified database structure. The proposed system is having further advantage of eliminating intermediate proxies that limit the scalability properties that are intrinsic in cloud-based system. The efficiency of the proposed system is evaluated through theoretical analysis and extensive experimental results based on a prototype implementation subject to the TPC-C standard benchmark for different numbers used in the distributed system.
The cloud can have both secured and unsecured files. We are focusing only the secured files and how we are going to secure that more efficiently. Data owner stored the files in cloud in encrypted format to avoid hacking. When the owner tries to download the data we have to provide more security like primary credentials, IP configurations, etc. The secured files should be simultaneously accessed by the original owners of that data. While the access of data the data should be concurrency and should avoid the unwanted more downloads of same file when the file has not been modified.

Our Scheme
This method describes how the sharing is done in the private cloud. Here due to K anonymity algorithm minute amount of can also be accessed, to overcome that problem here we go for the RSA algorithm, where a public key will generated which gives security even to that minute data.

1. Accountability
Accountability is the process of creating an account for new user's in cloud computing. Already existing user can login and access data which has been stored in cloud computing, such as data upload & data download. New user should register and create an new account in cloud computing for data access which has been stored in it. The new user's information will be stored and maintained by CIA (Cloud Information Administrator). Suppose, if the new user or an existing user have entered password wrongly, then the numerical method will be appeared. This image may contains an alphabet or with a combination of special characters or numeric. The user should enter the data which is displayed in that image, in order to avoid intruders.

2. MAC Address Tracking
Existing user's MAC address will be stored in CIA (Cloud Information Administrator). If an unauthorized person is trying to access the existing user's account, automatically a mail will be send to their mail-id with generates a password. And also, if an existing user is trying to access from different MAC address, automatically a mail will be send to their mail-id with generates a password.

3. Data Access and sharing

User can upload files in cloud storage. They can access their files which is stored in cloud storage by sharing with their files in another system. And also download it by enter the transaction password. They can able to view the history of the data used, and able to check the IP address used in the last five histories. User can check the number of files and memory of the backup files.

4. Data security

The existing user must enter the password to download their own files from cloud storage. To keep their files very secured in cloud storage, then it should be encrypted before it has been upload. Because, there might be a chance of accessing the user's files by their service provider without any logical process. So, in order to avoid an unauthorized (service provider) accessing of files, it should be encrypted by the owner of that files (i.e. User) before it has been upload in cloud storage. If the user want to access the files which is uploaded in encrypted form, have to download it after it has been decrypted.

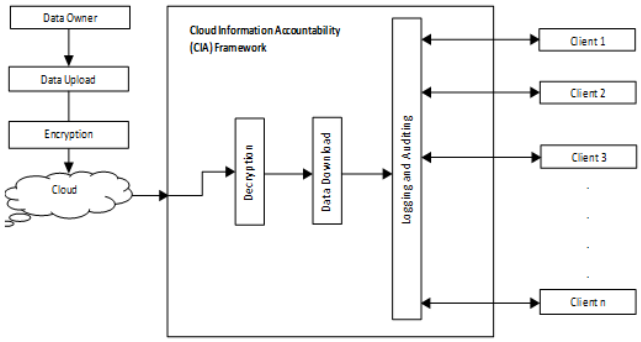

Fig: Architecture diagram

Anonymization Algorithm

A. Possible Solutions

The solution space is the set of all possible generalization rules, as in the classic k-anonymity. However, the set of accepted solutions which satisfy our guarantee is much greater. The problem of optimal multidimensional k-anonymity was proven to be NP-hard. In the worst case, i.e., an aggregate function f which takes a different value for every combination of the record attributes, the problem is the same. To deal with the complexity of the optimal anonymization problem we have opted for a heuristic solution. We cluster the records into equivalence classes, and perform recoding for the available resources of each class separately.

B. Algorithm

We propose a local-recoding generalization algorithm. As shown in the pseudo-code, our method has two main phases. Phase one divides the records into groups. We form equivalence classes with respect to the f function. First, all records are sorted with reference to their f(qi2, qi2, ..., qin) value. Then, they are clustered into equivalence classes of sizes $k \leq |EC| \leq 2k - 1$. We limit the EC size to avoid over generalizing values. In the second phase we consider each equivalence class separately, and perform generalizations to its values. If all records in a class EC already have the same result by the attacker using the rest of the values.

The basic idea is to form groups of records that have similar aggregate function values of their identifiers. To overcome this we perform local generalizations independently within the group. We reduce the process to numerical values, but our method is been extended to categorize the aggregate functions that are defined over them.

Input: D {Original dataset}, f{ Aggregate function}
K {privacy parameter}
Output: D* {kf -anonymous Dataset.}
1: for all tuples t < qi1, qi2, ..., qin >D do
2: estimate f(qi1, qi2, ..., qin)
3: sort according with reference to their f values.
4: form groups of size $\geq$ k and $\leq 2k - 1$
5: for every group EC do
6: if all tuples have the same f value then
7: add EC to D*.
8: else
9: Q = {QI1, QI2, ..., QIn} //Q contains all attributes
10: j = n − 1
11: while Q not empty do
12: estimate f for all combinations of j attributes
13: Let Cj be the combination with most similar f for all tuples
14: generalize the remaining attribute QIj = Q\Cj to a common range [vmin, vmax]
15: remove QIj from Q
16: j = j − 1
17: estimate f for all tuples in EC
18: if all tuples have the same f value then
19: break
20: add EC to D*
21: return D*
Algorithm1: k-anonymity

Experimental Evaluation

We evaluated experimentally the aggr Anon on real datasets from the UCI repository. The implementation was done in C++ and the experiments were performed on a Core i7 at 2.13GHz with 8GB RAM, running Ubuntu Linux.

Algorithms:

We compare our algorithm to Mondrian a multidimensional local recoding generalization algorithm for k-anonymity. The code we used is from, implemented in java. Experimental results suggest that aggrAnon preserves greater data utility than Mondrian, when considering attackers whose knowledge is limited to an aggregate function containing values. The aggregate function that we include in our process is sum (). The implementation is different and we do not compare the two algorithms in terms of execution time.

The k-anonymity algorithm is them compared with RSA algorithm which is simple and easy for the users, where the key will be generated and that will be used for the data access purpose. Even the minute date can only be accessed by the key generated through this algorithm.

A. Key Generation

RSA involves a public key and a private key. The public key can be known and is used for encrypting messages. Encrypted messages with the public key can only be decrypted in a particular amount of time with the help of private key. The RSA algorithm generates key in the following way:

1. Choose two distinct prime numbers p and q.
2. Compute n = pq.
3. Compute $\varphi(n) = \varphi(p)\varphi(q) = (p - 1)(q - 1)$, where φ is Euler's totient function.
4. Choose an integer e such that $1 < e < \varphi(n)$, $\gcd(e, \varphi(n)) = 1$; i.e., e ($\varphi(n)$) are coprime
5. Determine d as $d \equiv e-1 \pmod{\varphi(n)}$;

Algorithm2: Key generation

B. Encryption

Person A transmits his/her public key to Person B, keeping his/her private key secret. When Person B wishes to send the message "HI" to Person A, he first converts HI to an integer such that $0 < m < n$ by using agreed upon reversible protocol known as a padding scheme. Person B computes, with Person A's public key information, c the cipher text corresponds to

$$c \equiv me \pmod{n}. \quad (1)$$

Person B now sends message "M" in cipher text, or c, to Person A.

C. Decryption

Person A recovers hi from c by using his/her private key d, which computes

$$m \equiv cd \pmod{n}. \quad (2)$$

Given hi, Person A can recover the original message "HI" by reversing the padding scheme.

Conclusion

We proposed innovative approaches for automatically logging any access to the data in the cloud together with an auditing and logging mechanism. Our process allows the data owner to not only audit his content but also enforce strong back-end protection if it is needed. One of the major features included in our work is that it enables the data owner to audit even those copies of its data that were made without his/her knowledge. In future, we try to refine our approach to verify the integrity of the JRE and the authentication needed. For example, we will go for testing process whether it is possible to leverage the notion of a secure JVM that is implemented by IBM. The research work is aiming to provide the proper resistant to Java based. Our goal is to design a relevant and more generic object-oriented approach to facilitate autonomous protection of travelling content. We would like to support a variety of security policy, like index policies for text files, use of control for executable, and generic accountability and provenance controls.

References

[1] Gourkhede, M.H. ; Dept. of Comput. Sci. & Eng., G.H. Raisoni Coll. of Eng., Nagpur, India; Theng, D.P.”Analysing Security and Privacy Management for Cloud Computing Environment” 7-9 April 2014.

[2] Hai Yu ; Lightning Protection Center of Hainan Province, Haikou, China ; Tinglong Zhang ; Yi Gao ; Xiaoqing Lao ”A study on characteristics of spatial and temporal distribution of cloud” 11-18 Oct. 2014.

[3] Pfeifer, T.; Tech. Univ. Berlin, Berlin, Germany; Covaci, S.”Active protection of patient data by reverse cloud approach” 9-12 Oct. 2013.

[4] Chih-Yung Chen; Dept. of Inf. Manage., St. John"s Univ., Taipei, Taiwan; Hsiang-Yi Tseng”An Exploration of the Optimization of Excutive Scheduling in the Cloud Computing” 26-29 March 2012.

[5] H. Park and K. Shim “Approximate algorithms for K -Anonimity” in SIGMOD, pp.67-78, 2007.

[6] M. Terrovities, N. Mamoulis and p. Kalnis “Privacy-preserving Anonymization of Set valued Data,”PVLDB, vol.1, no 1, 2008.

[7] K. LeFere, D.J. DeWitt and R. Ramakrishnan, “Mondrian Multidimensional K-Anonymity,” in ICDE, 2006.