International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:1, 2014

Enhancing Privacy-Preserving Cloud Database Querying by Preventing Brute Force Attacks

Ambika Vishal Pawar, Ajay Dani

***Abstract*—**Considering the complexities involved in Cloud computing, there are still plenty of issues that affect the privacy of data in cloud environment. Unless these problems get solved, we think that the problem of preserving privacy in cloud databases is still open. In tokenization and homomorphic cryptography based solutions for privacy preserving cloud database querying, there is possibility that by colluding with service provider adversary may run brute force attacks that will reveal the attribute values.

In this paper we propose a solution by defining the variant of K – means clustering algorithm that effectively detects such brute force attacks and enhances privacy of cloud database querying by preventing this attacks.

***Keywords*—**Privacy, Database, Cloud Computing, Clustering, K-means, Cryptography.

I. Introduction

TO reduce computing costs and to improve work productivity enterprises across the world are looking at technology as a valuable partner. The recent advancements in cloud computing technology help enterprises in reducing the computing costs while boosting work productivity. With growing enterprise interest in cloud as a computing platform and rise of Software as a Service (SaaS) based applications, the demand for cloud technologies from enterprises is picking up. Apart from this with growing influence of web 2.0 technologies, non-enterprise users are also heavily using cloud technologies. In coming decades cloud technologies are going to become pillar for world trade.

Despite of cloud technology's enormous potential, there are still many technical issues that continue to hamper the wide spread adaptation of technology. [1] Recent Privacy threats experienced by users of services offered by Apple Inc. Google Inc., Amazon Inc. [1] are clear indications that cloud is intrinsically insecure from a user's viewpoint.

Because users don't have access to cloud service provider's internal operations, preserving privacy of user in cloud environment is a challenge for researchers. Fig. 1 describes a typical cloud-computing scenario; an enterprise user is using SaaS cloud services.

Based on Fig. 1, Yanbin Lu and Gene Tsudik [2] list following challenges in preserving privacy in cloud.

Ambika Vishal Pawar is Research scholar in Symbiosis International University, Pune, India (Mobile: 9921001033; e-mail: ambikap@sitpune.edu.in).

Dr. Ajay Dani is with Symbiosis International University (Mobile: 9423035761; e-mail: ardani_123@rediffmail.com).

- Challenge 1: While using SaaS applications on cloud platform enterprises outsource their data to cloud server? How to protect theft of such outsourced data?
- Challenge 2: How to protect such data from abuse by the cloud server?
- Challenge 3: How to design and implement content level fine-grained access control for users?
- Challenge 4: How to execute query on the database which is stored in cloud storage, without disclosing the query contents?
- Challenge 5: How to query contents from non-trusted entities?

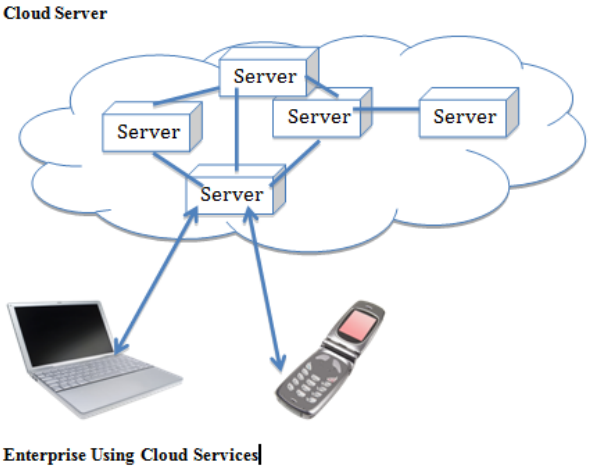

Fig. 1 SaaS Based cloud environment

Till now Lot of work has been done to preserve the privacy in cloud databases. Siani Pearson, Yun Shen and Miranda Mowbray [3] proposed a privacy manager for cloud computing. Jian Wang [4] proposed anonymity based method for privacy preserving. Miao Zhou [5] proposed a method for improved key management for preserving privacy. Qin Liu [6] proposed bilinear graph based privacy preserving keyword searching scheme for retrieving files. Haibo Hu [7] proposed comprehensive privacy preserving scheme for indexed data. Marten van Dijk and Ari Juels [8] argued that alone cryptography is not sufficient to preserve the privacy in cloud computing. Dr. Alexander Benlian [9] discussed about importance and adoption of SaaS. Shucheng Yu [10] proposed Fine grained Data Access Control in Cloud Computing using attribute based encryption. Although many schemes have been

proposed to preserve the privacy in cloud computing, most of them have failed to resolve all the above listed challenges.

Schemes proposed in [3] and [4] fail to consider comprehensive approach while solving the privacy preserving problem and do not provide solution for challenges 3, 4, 5 listed above.

Adi Shamir [11] in his break-through work first time proposed a novel type of cryptographic scheme called 'Identity-Based Cryptosystems and Signature Schemes' which enables any pair of users to communicate securely and to verify each other's signatures without exchanging private or public keys, with-out keeping key directories and also without using any third party services. It allows for a sender to encrypt a message to an identity without access to a public key certificate.

Amit Sahai and Brent Waters [12] proposed fuzzy identity based encryption scheme which they termed as 'Attribute - Based Encryption' (ABE). This scheme based on Adi Shamir [11] it allows for error tolerance between identity of private key and the public key used to encrypt a cipher text. In this scheme user's keys and cipher texts are labeled with set of attributes and user with secret key will be able to decrypt cipher text encrypted with public key if and only if they are within a certain distance of each other when measured by some metric.

Based on [12], Vipul Goyal et al. [13] proposed a new cryptosystem which they called 'Key -Policy Attribute-Based Encryption (KP-ABE)'. This system first time implemented fine gained access control on sensitive data that is shared and stored by third-party sites on the Internet. In this system cipher texts are labeled with sets of attributes and private keys are associated with access structures that control which cipher texts a user is able to decrypt. Bethencourt et al. [14] proposed a new cryptosystem called as 'Ciphertext-Policy Attribute-Based Encryption' (CP-ABE) which allowed user to store encrypted data to be kept confidential even if server on which data is stored is untrusted.

Lu and Tsudik [2] proposed a new cryptosystem for preserving privacy in cloud database querying. Based on [11]-[14] Lu and Tsudik elegantly solved all the previously mentioned challenges in the privacy preservation of the cloud. Moreover other attempts [6], [7] tried to solve the problem of privacy preservation in Cloud environment but they were directed at file and indexed data. Lu and Tsudik [2] have solution for relational databases which are widely used by SaaS based applications in the cloud.

Although Lu and Tsudik [2] have proposed novel scheme, this scheme still has following limitations.

- It hides the attribute values in conditional expression but access structure is revealed to the adversary.
- Join operations between two tables are not supported
- If set of attribute values in access structure is small, then adversary can always encrypt something under all possible values and can collude with cloud service provider to check match for these encrypted values. This might reveal attribute values in token.
- Enterprise server is required to be online continuously so that user can extract decryption keys and search tokens.

Unless these above mentioned limitations are overcome, we consider the problem of preserving data privacy in cloud based applications is still open.

In this paper we aim to solve the attribute value expose problem in cloud database querying when user and cloud database provider collude with each other.

II. Problem Definition

Fig. 2 shows Cloud storage architecture for typical SaaS based application. There are four entities.

- The Cloud Server (S)
- The database Owner(DO)
- Certification Authority (CA)
- User (U)

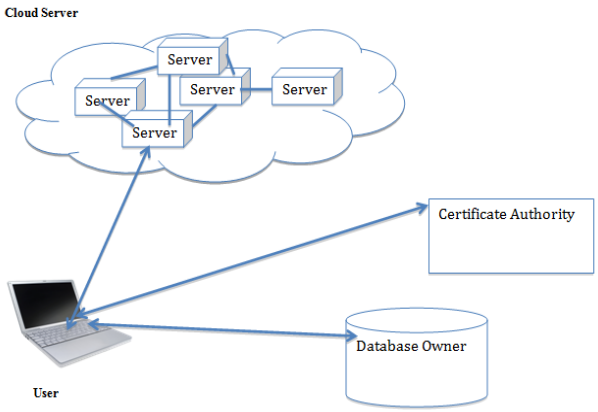

Fig. 2 Cloud Storage Architecture

The privacy preserving solution for cloud database querying proposed by Lu and Tsudik [2] works as follows: DO's database table consists of w attributes $\{\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_w\}$. Let $\Omega = \{1, 2, \ldots, w\}$

Let $\{v_i\}_{1 \le i \le w}$ a set of w values for each record m with each v_i corresponding to attribute α_i.

Step 1. Before starting DO runs ***Setup*** algorithm to initialize some parameters *params* and DO's master key msk_{DO}.

Step 2. DO takes *params* and msk_{DO} as input and encrypts all m records in the database by executing ***Encrypt*** algorithm and transfers them to Cloud server S offline. DO can encrypt new items later.

Step 3. Whenever user U wants to retrieve some data from cloud database S, it first forms a corresponding SQL query with monotonic access structure $\tau\Upsilon$.

Let $\tau\Upsilon$ be access tree constructed over a subset Υ of Ω and let $v\Upsilon$ be the set of values for $\tau\Upsilon$. A complete record can be viewed as $v\Omega$.

User U on input ($params, \tau\Upsilon, v\Upsilon$) and DO on input (params, msk_{DO}) engage in communication using interactive protocol. At the end of this communication U outputs a search token *tk*

(τϒ, vϒ) and a decryption key sk (τϒ, vϒ). DO outputs (τϒ, vϒ).

Step 4. User sends search token *tk* (τϒ, vϒ) to cloud server S who runs ***Test*** algorithm in order to find the matching records. S on input parameters *params*, a search token *tk* (τϒ, vϒ) and a cipher text

C = ***Encrypt*** (msk_{DO}, v'_{Ω}), outputs "yes" if τϒ. (vϒ, , v'Ω) = 1 and "no" otherwise.

Step 5. User U selects set of possible attribute values in γ that is small and tries to encrypt all possible values under that attribute. U then asks S to run ***Test***

Over the encryptions to see if there is match. Thus step 3 and 4 get executed repeatedly.

As set of possible attribute values is small, Cloud service provider runs brute force attack. This reveals $v\Upsilon$ within *tk* (τϒ, vϒ). This leads to loss of privacy of user U.

We have to prevent this loss of privacy of user U when he is querying cloud database in step 5. To prevent this possible loss of privacy first we need to consider challenges.

Challenge 1: As set of possible attribute is very small and access structure τϒ is known to cloud service provider S, S can always guess all possible values for that attribute and by running brute force attack, vϒ will be revealed. For example if the database is related with healthcare there could be attribute like age whose set possible values is very small resulting in revealing vϒ. The simple solution is hiding the access structure τϒ so that S will not come to know what attribute user U is searching. But even If we successfully manage to hide the access structure τ_{Υ} this will not solve the problem comprehensively. As S is a SaaS service provider, he is very well aware with database attributes and he can always read the pattern in attribute values and can guess the attribute.

Challenge 2: There may a SaaS based application where S knows the attribute name but has no idea about the set of possible attribute values. For example in a Customer Relationship Management (CRM) software there may be attributes whose values cannot be guessed beforehand but has small possible set of values. For example CRM SaaS application may have attributes like annual income of prospects, budget he has allocated for buying new product, his educational details etc. These attributes have small set of possible values and S cannot guess their values beforehand but can do so later when there is sufficient number of values in database corresponding to these attributes. He can analyze the pattern and then run brute force attack that will reveal vϒ .

III. Our Contribution

In this paper we propose a system that will instead of detecting, prevent the brute force attack that has been described in problem definition section. We propose system architecture for preserving data and data querying privacy in SaaS based applications in cloud. We design a variant of K-means algorithm for clustering the attribute values that forms basis for further processing. We design this variant of K-means algorithm by considering various issues and constraints that are specific to our problem domain. We then modify the algorithms in [2] which prevent the possibility of brute force attack by cloud service provider.

IV. Related Work

The problem has been discussed in multiple communities such as the database community, the cryptography community as well as in the statistical disclosure control community. The key directions in the field of privacy preserving in cloud computing is as follows:

A. Privacy Preserving Data Publication

Techniques proposed in [15]-[18] consider privacy preservation problem only while data publication. These privacy-preserving techniques include randomization [15], k-anonymity [16], [17] and l-diversity [18].

Data retrieval and data publication both are critical operations in any cloud computing. As above-mentioned techniques are restricted with privacy preservation only during data publication and are of little use in cloud computing. Our work is based on [2] which consider privacy preserving essential during data retrieval as well as data publication.

B. Privacy Preservation of Data Mining

Techniques mentioned in [19], [20] and many such techniques are concerned with privacy preserving issues in data mining.

Whereas our work is primarily concerned with privacy preservation issues in Software as a Service (SaaS) based applications in cloud computing.

C. Privacy Preservation of Non RDBMS Data

Techniques proposed in [6], [7] are concerned with privacy preservation in cloud computing but consider non RDBMS data for their problem.

Our work is solely specific to SaaS based RDBMS data in cloud computing.

D. Distributed Privacy Preservation

Reference [21] contains methods of cryptography in distributed privacy preservation.

Our work is solely specific to SaaS based RDBMS data in cloud computing.

E. Clustering Algorithms

Various algorithms have been proposed to for data clustering. Reference [22] proposes K-means algorithm for clustering where K number of data clusters are formed and centroids are updated in iterative manner. Reference [23] deals with problem of data clustering with fuzzy K-means algorithm which modifies K-means in [22]. A novel minimum spanning tree based algorithm has been proposed in [24] for data clustering. Reference [25] proposes mutual neighborhood method for data clustering. References [26] and [27] propose single link and complete link based data clustering methods respectively.

Our work is based on through analysis of problem domain where after comprehensive analysis of problem domain we argue that of all the methods available for data clustering K-means can be extended to solve the problem that we have defined in section II of this paper. Moreover our scheme substantially modifies K-means to effectively solve the problem. We use K-means based data clustering to group close values together so that many unnecessary comparisons can be eliminated.

F. Deciding Values of K in K-Means Algorithm

Success of K-means algorithm depends upon careful selection of value K. Reference [28] decides the value of by density estimation of data at various locations. Reference [29] proposes modifications to two highly successful hierarchical initialization methods namely Var-Part and PCA-Part and employs a discriminant analysis based approach to select value of K. Reference [30] proposes a methods that selects value of K by using the information obtained in clustering operation itself. Reference [31] improves the K-means clustering algorithm by using statistical test for deciding the value of K.

It is clearly evident that most of the work that has been done to find the value of K has been very specific and cannot be applied in general to all the problems. There are several factors like type of data, data density etc. that affect the value of K. Hence in our method we proposed our own value of K that is specific to the problem domain we are working in. According to [9] SaaS based cloud applications are used by small and medium businesses as well as large enterprises. Reference [9] also lists type of applications that are popular in SaaS based cloud model. This compels us to argue that the value of K should be different for small and medium business users than large enterprises. This is because generally small users may have database records in few thousands whereas for large enterprises this value may be much larger. In our work we restrict ourselves to small and medium business users and decide the value of K by our K selection algorithm, which is very specific to our problem definition.

G. Efficient Clustering Techniques

Another factor that is crucial for successful clustering is how actually clusters are formed in iterative manner. Reference [32] employs a statistical technique called Principal Component Analysis (PCA) for efficient unsupervised data clustering. Reference [33] shows that different K-means algorithm do behave differently from each other on simple low dimensional synthetic datasets. Reference [34] improves clustering substantially by able to identifying non spherical clusters. Reference [35] shows how clustering can be improved by exploiting sparsity of the data set.

From all above-mentioned methods we can safely say that after initialization, the continual progress and subsequent convergence of any clustering algorithm depends on several factors. Again several parameters like type of data, data density become important in designing efficient clustering algorithm. In our solution we mainly want to detect the repetition of numbers over small value space. We cluster the data values that are close and then detect whether they are getting repeated over small set. Instead of clustering a simple solution could have been an efficient algorithm for comparing data values. Clustering of data values groups close data values together and substantially reduces number of comparisons. This scale well when number of data records in the database increase exponentially.

H. Encryption in Cloud Databases

We modify the encryption algorithms in [2] to prevent any guessing of attribute values being searched. We do so by obfuscation of vulnerable data using index tables. This retains searching ability. The privacy achieved outweighs the extra computational overhead generated during generating extra tokens.

V. Our Solution

Considering the related work that has been done to solve the problem of privacy preservation in cloud computing generally and SaaS based applications specifically.

Fig. 3 illustrates detail design of our system. We consider following design issues for any solution that aims to solve the said problem holistically and comprehensively

- Any new scheme that is proposed should execute its critical tasks in user environment instead of cloud environment.
- Such scheme would create minimum possible resource overhead for user.
- Such scheme should get integrated seamlessly with a web based SaaS application as ***Test*** phase of solution proposed by Lu and Tsudik [2] will be implemented as a part of SaaS application.
- There are plenty of issues that need to be solved to achieve preservation of privacy in SaaS based applications and hence such scheme should be a part of larger system architecture where it will get integrated and scaled easily.

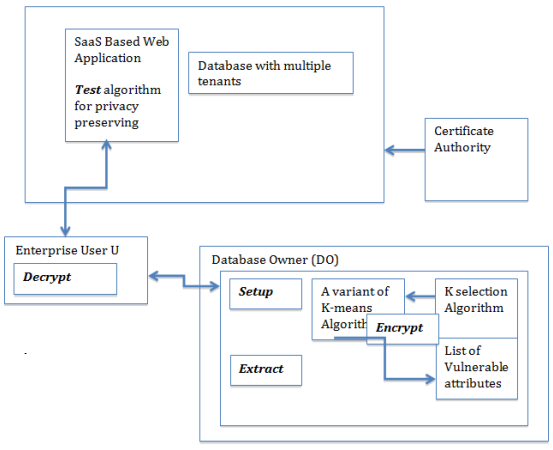

Fig. 3 System Design

Our solution of preventing brute force attacks will be part of system architecture proposed below.

The solution proposed by Shiyuan Wang et al. [36] tries to solve the problem from database side and fails to address User U's concerns about his privacy protection as critical tasks that protect privacy still get executed in cloud environment.

A. Our Scheme

Considering the challenges that we have listed above while preserving privacy in SaaS based applications, below is our scheme that prevents brute force attack that reveals attribute values

Step 1. Prepare the list of attributes $\{\alpha_1, \alpha_2, \alpha_3, ..., \alpha_w\}$ and sequentially process the list by running a variant of K-means algorithm described in section B.

Step 2. Prepare the list of attributes that are labeled as ***vulnerable*** by a variant of K-means algorithm.

Step 3. Run the privacy preserving algorithm described in subsequent section D.

Example

Let input attribute set is $\{\alpha_1, \alpha_2, \alpha_3, ..., \alpha_w\}$ = {Emp_Name, Emp_Age, Emp_Address, Emp_Income) for employee details table

Step 1. We will process each of the attribute in given list sequentially by applying variant of K-means as stated in section B i.e. first we will check for attribute Emp_Name then Emp_Age and so on for their value patterns whether they have got repeated values.

e.g. Emp_Age attribute has many repeated values. Let's assume there are total 10 records in database with these age values 28, 28, 32, 32, 45, 45, 55, 55, 57, 58. Out of 10 only 2 values are nonrepeated. Emp_Age Attribute will be vulnerable as more than 80% values get repeated.

Repetition ratio=Repeated values/Total values=8/10=80%

Similarly we define non repetition ration as

Non repetition ratio= Non repeated values / Total values =2/10=20%

Step 2. Based on the result of step 1 we will prepare list of vulnerable attributes.

Step 3. For vulnerable attributes we will apply technique in section D.

B. A variant of K-means Algorithm

Let K be the number of clusters.

Let C_i. be the i^{th} cluster.

Let c_i be the centroid of the cluster C_i.

Let x_i be the data object.

Step 1. Initialize the K means algorithm by selecting K as given by K selection algorithm.

Step 2. For the selected attribute α, select random K number of data objects from the set of attribute values and initialize them as centroids.

Step 3. Assign each data object to its nearest centroid by using below mentioned proximity function. Data object should be assigned to centroid that for which it has lowest proximity value.

For every data object x_i find the proximity value P_i

P_i = dist (c_i, x_i)

Step 4. From the cluster remove the data objects x_i, for which value of P_i is 0 and also remove the c_i, corresponding centroid from the cluster.

Step 5. Repeat step 2, 3 and 4 for remaining data objects until there are no data objects left or there are centroids for whom there are data objects that have non zero p_i

Step 6. If no data objects are left or ratio of data objects left to total no. of data objects is less than 0.40 then label the set of attribute values as ***vulnerable*** else label as normal attribute.

Step 7. For ***vulnerable*** attributes prepare the list of all previous centroids i.e. set of repeated data objects and label it as ***vulnerable data objects list.***

Example:

Let us consider for 20 no. of records with Emp_Age = {21, 25, 28, 30, 32, 40, 45, 50, 55, 60, 25, 28, 30, 32, 40, 45, 50, 55, 28, 30}

Let us apply variant of K-means algorithm to this set of values

Let K=3 as in step 1 no. of records<3000

Let us randomly select cetroids, c1=25 for cluster C_1, c_2=40 for cluster C_2, c3=55 for cluster C_3 as given in step 2

Let us form clusters as explained in step 3

P_1 = dist (c_1, x_1) = dist (25, 21) = 3

P_2 = dist (c_2, x_1) = dist (40, 21) = 18

P_3 = dist (c_3, x_1) = dist (55, 21) = 34

So 21 will be placed in C_1, as it is nearest cetroid is c_1.

Similarly we will place all values in given set to respective cluster.

Resultant clusters will be formed as shown below in Fig. 4

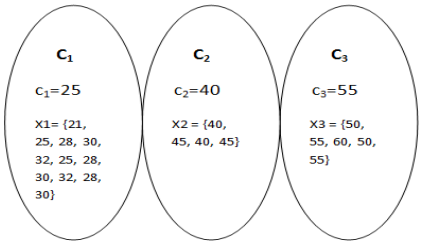

Fig. 4 Variant of K-means example

Now we will remove 25, 40 and 55 as these values are repeated in C_1, C_2 and C_3 respectively.

Now set of remaining data objects=

{21,28,30,32,28,30,32,28,32,45,45,50,60,50}

Recursively we will apply same procedure for remaining data objects. At the end of this process we will get remaining data objects which are not repeated. We will find out non repetition ratio and will declare Emp_Age vulnerable attribute, if this ratio come less than 40%.

C. K- Selection Algorithm

Step 1. Find the total number of records m in the database.

Step 2. If (m<3000)

K= 3 else

K=m /1000.
Step 3. End

D. Enhancing Privacy Preservation

The problem of detecting and preventing possibility of taken data leakage is an essentially the problem of pattern detection. Above mentioned k-means variant successfully detects the pattern in which database values get repeated extensively over small data set. However such data leakage can easily be prevented by simple technique of inserting dummy records in database which will ensure that no pattern is formed in database. Subsequently periodic retrieval of these dummy records using ***Extract*** and ***Decrypt*** in [2] will strengthen the data leakage prevention. An illusion created by DO by presenting dummy data as real data to Cloud service provider tricks cloud service provider by breaking any data pattern formation.

VI. Conclusion

This paper solves the critical problem of preventing brute force attack used for guessing the attribute values user is searching. This is one step towards preserving privacy of user while using SaaS based applications in cloud environment in comprehensive manner.

Our framework and various algorithms that are part of this architecture successfully prevents possibility of such brute force attack but tradeoff between accuracy and computational overhead has to be made as to protect privacy we introduce new computations. Further while preventing brute force attack we have used 'look up table' based approach to preserve privacy. Although this does preserves the searching operations, however to preserve complex mathematical operations where we store the values obtained from mathematical operations of encrypted data better tokenization look up table system has to be developed. However ability of preserving such mathematical operations will change for every SaaS based application. SaaS based application provider can design his mathematical operations preserving technique specific to his application. Further we expect that all the algorithms proposed in our system can be optimized and improved to offer better system.

References

[1] Kui Ren, Cong Wang, and Qian Wang, Security Challenges for the Public Cloud, Internet Computing, IEEE (Volume: 16, Issue: 1).
[2] Y. Lu and G. Tsudik, Privacy-Preserving Cloud Database Querying, Journal of Internet Services and Information Security (JISIS), Vol. 1 No. 4, November 2011.
[3] Siani Pearson, Yun Shen, Miranda Mowbray, A Privacy Manager for Cloud Computing, First International Conference, CloudCom 2009, Beijing, China, December 1-4, 2009. Proceedings.
[4] Jian Wang, Yan Zhao; Shuo Jiang; Jiajin Le, Providing privacy preserving in cloud computing, International Conference on Test and Measurement, 2009. 213-216.
[5] Miao Zhou, Yi Mu, Willy Susilo Jun Yan, Liju Dong, Privacy enhanced data outsourcing in the cloud, Journal of Network and Computer Applications, 35 (2012) 1367–1373.
[6] Qin Liu, Guojun Wang, Jie Wu, Secure and privacy preserving keyword searching for cloud storage services, Journal of Network and Computer Applications 35 (2012) 927–933.
[7] Haibo Hu; Jianliang Xu; Chushi Ren; Byron Choi, Processing private queries over untrusted data cloud through privacy homomorphism , IEEE 27th International Conference on Data Engineering (ICDE), 2011.
[8] Marten Van Dijk, Ari Juels, On the Impossibility of Cryptography Alone for Privacy-Preserving Cloud Computing.
[9] Dr. Alexander Benlian, Prof. Dr. Thomas Hess, Prof. Dr. Peter Buxmann, Drivers of SaaS-Adoption – An Empirical Study of Different Application Type, Business & Information Systems Engineering October 2009, Volume 1, Issue 5, pp 357-369.
[10] Shucheng Yu Cong Wang ; Kui Ren ; Wenjing Lou, Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing, INFOCOM, 2010 Proceedings IEEE.
[11] Adi Shamir, Identity-Based Cryptosystems and Signature Schemes, Proceedings of CRYPTO 84.
[12] Amit Sahai, Brent Waters, Fuzzy Identity-Based Encryption, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005. Proceedings.
[13] Vipul Goyal,Omkant Pandey, Amit Sahai, Brent Waters , Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data, Proceeding CCS '06 Proceedings of the 13th ACM conference on Computer and communications security Pages 89 - 98 ,ACM New York, NY.
[14] John Bethencourt, Amit Sahai, Brent Waters, Ciphertext-Policy Attribute-Based Encryption, IEEE Symposium on Security and Privacy 2007 (SP' 2007)
[15] Agrawal R., Srikant R. Privacy-Preserving Data Mining. ACM SIGMOD Conference, 2000.
[16] Samarati P., SweeneyL. Protecting Privacy when Disclosing Information: k-Anonymity and its Enforcement Through Generalization and Suppression. IEEE Symp. on Security and Privacy, 1998.
[17] Bayardo R. J., Agrawal R. Data Privacy through optimal k-anonymization. ICDE Conference, 2005.
[18] Machanavajjhala A., Gehrke J., Kifer D. l-diversity: Privacy beyond k-anonymity. IEEE ICDE Conference, 2006.
[19] Aggarwal C. C., Yu P. S.: A Condensation approach to privacy preserving data mining. EDBT Conference, 2004.
[20] Aggarwal C. C., Yu P. S.: On Variable Constraints in Privacy-Preserving Data Mining. SIAM Conference, 2005.
[21] Pinkas B.: Cryptographic Techniques for Privacy-Preserving Data Mining. ACM SIGKDD Explorations, 4(2), 2002.
[22] A. Hartigan and M. A. Wong Algorithm AS 136: A K-Means Clustering Algorithm Journal of the Royal Statistical Society. Series C (Applied Statistics) Vol. 28, No. 1 (1979), pp. 100-108.
[23] Mark Junjie Li, Michael K. Ng, Yiu-ming Cheung: Agglomerative Fuzzy K-Means Clustering Algorithm with Selection of Number of Clusters, IEEE Transactions on Knowledge and Data Engineering, Vol. 20 No. 11, November 2008.
[24] O. Grayorash, Y. Zhou, and Z, Jorgenssn, Minimum Spanning tree based clustering algorithms, Proc,. of IEEE Inn. Conf Tools with Artificial Intelligence, pp 73-81, 2006.
[25] K. Chidananda Gowda and G. Krishna The Condensed Nearest Neighbor Rule Using the Concept of Mutual Nearest Neighborhood, IEEE Transactions on Information Theory, vol. It-25,no. 4, july 1979.
[26] R. Sibson, SLINK: An optimally efficient algorithm for the single-link cluster method. The Computer Journal, Volume 16, 1973.
[27] D. Defays. An efficient algorithm for a complete link method, The Computer Journal (1977).
[28] Stephen Redmond, Conor Heneghan , A method for initialising the K-means clustering algorithm using kd-trees, ACM Journal of, Pattern Recognition, Volume 28 Issue 8, June, 2007 Pages 965-973.
[29] M. Emre Celebi, , Hassan A. Kingravi , Deterministic Initialization of the K-Means Algorithm Using Hierarchical Clustering, International Journal of Pattern Recognition and Artificial Intelligence, 26(7): 1250018, 2012.
[30] D T Pham, S S Dimov, and C D Nguyen ,' Selection of K in K-means clustering', Proc. IMechE Vol. 219 Part C: J. Mechanical Engineering Science.

[31] Sebastian Thrun, Lawrence K. Saul 'Learning the k in k-means Greg Hamerly' Advances in Neural Information Processing Systems, Volume 16.
[32] Chris Ding, Xiaofeng He, K-means Clustering via Principal Component Analysis, Proceedings of the 21 st International Confer- ence on Machine Learning, Banff, Canada, 2004.
[33] Greg Hamerly, Charles Elkan , Alternatives to the k-means algorithm that find better clusterings, ACM Proceeding CIKM '02 Proceedings of the eleventh international conference on Information and knowledge management, Pages 600-607.
[34] Guha S., Rastogi R., Shim K. CURE: An efficient clustering algorithm for large databases, Year: 1998 ACM, Source title: SIGMOD Record Volume: 27 Issue: 2 Page: 73-84.
[35] Inderjit S. Dhillon, James Fan , Yuqiang Guan , Efficient Clustering of Very Large Document Collections, Data Mining for Scientific and Engineering Applications, Springer, 31-Oct-2001.
[36] Shiyuan Wang, Divyakant Agrawal, and Amr El Abbadi, A Comprehensive Framework for Secure Query Processing on Relational Data in the Cloud, Proceeding SDM'11 Proceedings of the 8th VLDB international conference on Secure data management Pages 52-69.

Ambika Pawar has completed her B.E and M.E in Computer Science and Engineering from PUNE University in 2002 and 2008. She is pursuing PHD from Symbiosis International University. She has ten years of teaching experience in Computer and IT Departments of Engineering Colleges. She has been an Assistant Professor in Computer Science and IT Department, Symbiosis Institute of Technology, since 2010. Her research interest includes Data Structures, Data Privacy, and Cloud Computing.

Dr. Ajay Dani has completed M.Tech in Computer Science from IIT Khadakpur and PHD in Computer Science and Engineering from Hyderabad Central University. He has total 20 yrs industry experience in Research and Development and 2 years of experience in Teaching. He has publications in seven international journals, 20 conferences. His research interests are Data Mining, Cloud computing, Data Privacy and Security, Databases, Distributed Computing.