Efficient Cloud Data Confidentiality for DaaS

Atiq ur Rehman, M.Hussain
SZABIST Islamabad Pakistan
hafizatiq@gmail.com, hmureed@yahoo.com

Abstract

Privacy is a pinnacle concern of the cloud database model known as "Database as a service (DaaS)". DaaS is highly appreciated in business community because it saves hardware cost, cost of the technical people required to manage the database and it also saves the license cost of the database. Moreover, it offers reliable services and people can access their data 24 x 7 from anywhere provided the internet connection is available. Despite of all these advantages enterprises are reluctant to adopt DaaS, because of two types of threats that are associated with it. Firstly, it can be attacked by the hacker and secondly the privacy of data can be compromised by the administrators, managing the cloud database environment. In this paper we have focused on the second issue and proposed a model to protect privacy of data stored in cloud databases.

As per proposed model we encrypt and obfuscate data on client side before sending to the cloud database. In addition we offer mechanism to query over encrypted and obfuscated data on server side. Once the required data is filtered on server side, it is transferred on client side where the de-obfuscation and decryption is performed. Experiment results are also highlighted showing the enhancement in performance due to obfuscation factor.

Keywords: *DaaS security, data confidentiality.*

1. Introduction

Cloud computing is the new IT model that works on pay per use mechanism. This model is very appealing because it saves IT investment cost, in terms of infrastructure, software licenses, IT skilled manpower etc. Cloud computing based solutions can be deployed in three different deployment forms, namely public, private and hybrid cloud. Furthermore, cloud provider can offer services by three types of delivery models and those are infrastructure as a service (IaaS), software as a service (SaaS) and platform as a service (PaaS). These delivery models don't delegate full control to the data owner instead the control on data is shared by the service provider. Every delivery model offers different levels of control, like IaaS offers maximum control and SaaS offers minimum control to the data owner. But none of the delivery models provide complete control to the data owner, this leads toward the lack of trust on cloud paradigm.

The management of data is very important aspect of companies. Enterprises Interested to get benefits of cloud paradigm, outsource their data to database service provider and access their data via internet. This data management model is referred as database as a service (DaaS). DaaS is one of the most important applications of SaaS delivery model. DaaS model offers many benefits to enterprises like it saves the cost of database administration, offers reliable storage and efficient processing of query over the powerful machines of cloud service provider. In DaaS model, companies store business critical data through internet into the database that is managed by the service provider's database administrators (DBA). Database administrators have to have full control to the database to perform responsibilities of DBA like

database backup, database restore, recovery of database in case it crashed and also to perform performance and tuning of the database. This situation leads toward two types of threats, one from the hacker and the other from the privileged user of service provider who have full access to the data. In this paper we have focused on the second threat which is from the privileged user of service provider.

Although DaaS model is attractive, however it is not successful because the DBA can look into the data and can transfer business sensitive information to the competitors. Traditional privacy preserving solution like encryption is not successful in DaaS model. As if we store the encrypted data into the database offered by service provider, then it has to decrypt on server side once the user makes query from the database. During this time the privileged user of service provider can look into the data and so the confidentiality can be compromised. Moreover some queries generate results after statistical analysis. For example a user might need to display customer's information who has purchased items of more than 2000$ in a month, so that a discount voucher could be offered to him. Such types of queries are hard to run on encrypted data, where mathematical / statistical analysis needs to perform.

Some researchers have used a technique called obfuscation (disguise). This technique is also not successful to adopt for complete confidentiality of data in DaaS model because user can find values through reverse engineering or by using brute force technique, which may compromise security.

In this paper we are proposing a model to preserve confidentiality for data stored in DaaS model. The proposed model stores sensitive data with a combination of encryption and obfuscation. The sensitive character based attributes become encrypt and numeric sensitive columns become obfuscate based on the metadata repository. Moreover query will be executed over encrypted or/and obfuscated data. Once the required data is filtered on provider end then it is transferred to client where it is decrypted and de-obfuscated before the presentation to the user.

The remaining paper is structured as follows. Section 2 discusses the security threats to cloud databases. Section 3 discusses the related work. Section 4 provides the detail of the proposed model. Section 5 discusses the results of experiment, section 6 concludes the paper and section 7 discusses the future work.

2. Security threats to DaaS

Enterprises are reluctant to adopt DaaS model because of the following main security threats.

- Privacy of data can be compromised as privileged user has control over the data.
- Integrity of data can be compromised because of parallel cloud provider's control to the data.
- Segregation of data because of multi tenancy characteristic of DaaS model.
- Location of data is also one of the main issues because every country has different type of privacy related laws.

Although every security threat is important to resolve however, in this paper we are focusing only on privacy issue.

3. Related Work

3.1 Executing SQL over encrypted data in the database-service-provider model

Hacigumus et al [1] have proposed a model to achieve confidentiality of database deployed in cloud paradigm. Normally database working on cloud paradigm has two types of threats, one from the outsiders/hackers and the second one from the database administrator/ privileged user who is managing the database environment. Data owner is reluctant to use database utilities available as cloud service because of these threats. However authors have focused on second threat where information leakage is due to privileged user. In this paper authors have proposed to store encrypted record into the database. Authors have also added indexes with encrypted relation. To smooth the progress of indexing they have used the support of bucketing information. In order to execute query over encrypted data this bucketing information is used and it returns the segment of genuine query result. According to proposed approach first data is encrypted at client side before sending to database and indexing information is also maintained as metadata on client side. Similarly when query is executed over encrypted data it returns the encrypted data on client side where first it is decrypted, secondly it is transformed and then displayed on terminal. Although this technique is very effective to achieve confidentiality, however this filtering technique is very expensive in terms of time, as it introduces latency in the transaction.

3.2 A client based privacy manager for cloud computing

Miranda et al, in [2] have proposed "user-centric" trust model to achieve data confidentiality of cloud paradigm. According to proposed model data owner control's the sensitivity of data through privacy manager which is available on client side. This privacy manager uses the technique called "obfuscation" to achieve confidentiality rather than the traditional encryption technique. According to the given approach the data is first disguised with mathematical functions or by using any logic on client side and then it is send to service provider end for storage. Authors have used the database of "salesforce.com" where enterprises upload their business critical information. According to the given scenario the security threat to data is from service provider end. The presented framework has five modules named "Preference", "Data Access", "Feedback", "Personae" and "obfuscation" module. Preference module allows data owner to set confidentiality related inclinations. Data access module stores log whenever confidentiality related violations observed. Feedback module informs about the use of information. Personae module allows the data owner to select different security levels during interaction with service provider. Obfuscation module defends confidentiality of data when service provider is malevolent. Although this technique is efficient, however confidentiality of data can be compromised through brute force attack.

3.3 Executing query over encrypted character string in databases

In this paper authors have focused to achieve confidentiality of data for DaaS model of cloud paradigm. Hong et al [3] have proposed a scheme to query efficiently over encrypted

character data stored into the database available as DaaS. In this paper authors have proposed characteristics matrix to elaborate the uniqueness of character string. This character string of sensitive attribute is then compressed into the bit string and considers it the index of the string character. This bit string is then stored in the index column of sensitive attribute. This index attribute is used to translate range queries and like queries of the sensitive attribute. Moreover authors have analyzed their results with respect to efficiency and security. Although the proposed model looks practical but it requires extra storage because it stores index record against each value stored into the database.

3.4 Storage and query over encrypted character and numeric data in database

In this paper Zheng et al [4] have proposed a model for outsourced databases to achieve privacy of the data. Authors have used novel approach to efficiently query over encrypted data. Authors have categorized the data types and used different approaches to store and query for different types of data to enhance the performance. For character type of data authors have proposed to add additional field which stores characteristic about the fields. This additional field helps to search the records without involving too much query conditions and hence improve the performance. For numeric type of data authors have proposed to create B+ tree before encryption. Because of this process an index will be created which includes numerical data and its row location. Again this process will lead to quick search of the record for numerical type of data. Although the proposed model improved efficiency however it requires changing at DBMS level. Moreover it also needs extra storage space to store information for index fields.

4. Proposed Model

4.1 Overall Architecture

The proposed model is given in fig 1. In order to store data into the database, first required data encrypt or/and obfuscate on client side according to the requirement through respective module. This privacy related information is stored on metadata repository available on client side. After achieving privacy of data on client side, data is transferred to the database available as DaaS on provider side.

Moreover query executes on encrypted or/and obfuscated data to preserve confidentiality. To achieve confidentiality data is not decrypted/ de-obfuscated on server side. When user queries from the database available on remote site, it is first transformed on client side to run on encrypted or/and obfuscated data. After the transformation, query is transferred on server to execute and fetch the required data from the database engine. When required data is fetched on server side then it is transferred to client side where it is first decrypted/de-obfuscated before presentation to user terminal.

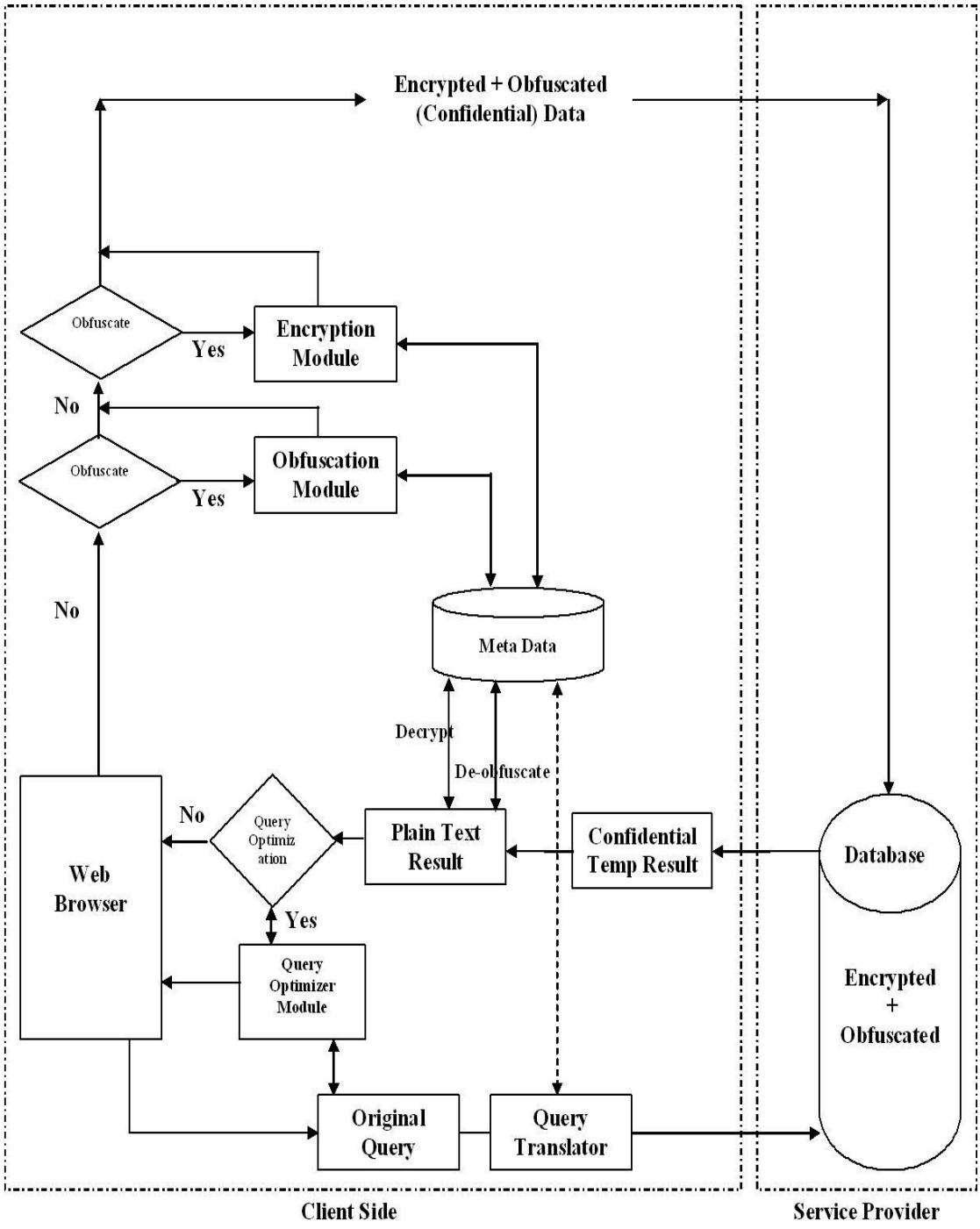

Fig 1: Architecture of Proposed Model

4.2 Storage mechanism

To better understand this model we have considered sales table including sample information given in table 1, to store on DaaS.

Table 1

Sale_date	Customer_Id	Prod_Name	Items	Price
02/11/2011	hafizatiq@gmail.com	Lipton	2	200
02/11/2011	h.mureed@yahoo.com	Tapal	3	150
02/11/2011	hafizatiq@hotmail.com	Tapal	4	200
02/12/2011	smbari@xyz.com	Lipton	3	300

As per our proposed model we don't encrypt or obfuscate complete record. Instead we encrypt only those character based sensitive attribute where statistics based conditions are not required during query, like customer_id and prod_name in our case. For this purpose any encryption algorithm can be used. Similarly we have proposed to obfuscate only those number based sensitive attribute where statistics based calculation is required like price in our case (Obfuscation is a process by which we disguise illegal users by applying through particular mathematical functions or by using through programming techniques). Rest of the columns like sale_date and items are stored as plain text, because these columns don't provide any useful information if we encrypt (Customer_Id, Prod_ Name) and obfuscate price columns as mentioned above. As we don't encrypt/ obfuscate every column so we don't need to store transformation keys un-necessarily on client side and even transformation mechanism will be more efficient during query, especially if "where" criteria of query involves only plain text columns.

4.3 Metadata Repository

Metadata repository is created on data owner side and it will store encryption/ decryption keys and obfuscation/ de-obfuscation information of required data. For example as per our case it stores encryption/ decryption keys for customer_id and prod_name and obfuscation/ de-obfuscation keys for price column as shown in table 2. This metadata repository is particularly used for transformation of queries. This transformation of query is necessary in order to preserve confidentiality. This metadata repository can be created in any text file, however it is recommended to store in small DBMS like MS-Access to achieve better performance.

Table 2

Column Name	Transformed Column Name	Type of Confidentiality	Key to lock	Key to unlock
Sale_date	00	Plain	-	-
Customer_Id	01	Encrypt	Ax675	B87cF
Prod_Name	02	Encrypt	Hy85	Gj7dH
Items	03	Plain	-	-
Price	04	Obfuscate	Under_root	squareroot

Moreover metadata repository also stores index keys for those encrypted columns where we need transformation. As per our scenario we have shown in Table 3.

Table 3

Prod_Name	Index key
Lipton	01
Tapal	02

4.4 Storage on Database Server

Information that user submits from client side as shown in table 1 is stored in encrypted/obfuscated form as shown in Table 4.

Table 4

00	01	02	03	04
02/11/2011	BQIcoQT0dH//ZHd8OEi/0Dk/gBu0Xfyt	CxgdZTNmmTA=	2	14.14
02/11/2011	xA7Axe4bmDQ=	xA7Axe4bmDQ=	3	12.24
02/11/2011	l2mfb2M2nkc=	xA7Axe4bmDQ=	4	14.14
02/12/2011	L3jknmb2M2nkc=	Ax89Axf4gmhgq=	3	17.32

It is obvious from above table that confidentiality can be preserved if we encrypt and obfuscate only sensitive columns.

4.5 Query Mechanism

When users query record from the database running on cloud provider end, his/her query first transformed to execute on encrypted or/and obfuscated data. This transformation is performed on client side based on the information stored in metadata repository.

After this transformation, query executes on encrypted or/and obfuscated data without decryption or de-obfuscation on server side. Once the required data is fetched, it will be transferred on client side for de-obfuscation as well as for decryption. This decryption or /and de-obfuscation is performed based on the information stored in metadata repository. Some queries may require further processing or filtering, in that case original query will be executed again on the temporary result available at this stage before sending to user terminal.

5. Experiment and Analysis

The objective of this simulation is to validate the performance of our proposed model. For this purpose we have created sales data table in database on server side as shown in Table 4. For transformation purpose we have created metadata table on client side as shown in table 2 and 3. To encrypt character based sensitive columns we have used DES algorithm. Similarly,

to obfuscate numeric type sensitive columns we have used square root function. The experiment is performed on Pentium 4 machine having speed 2.4 GHZ and 3 GB of RAM. Moreover related software used are, windows XP as the operating system, visual studio.net for development of simulation software and SQL server as a database on server side. In addition we have used Microsoft Access for metadata repository on client side.

5.1 Experiment 1 (record insertion)

In the first series of experimentation, we performed test for data storage and calculated time separately on client side as well as on server side for three different scenarios. In first scenario we send plain data (without encryption and obfuscation) to the database server available at service provider end for storage. In second scenario we used encryption for all sensitive columns on client side and therafter encrypted data is transferred to the database server available at service provider's end for storage. In third scenario we used encryption technique for character based sensitive columns and also used obfuscation technique for numeric type sensitive columns on client side. Once the data is encrypted and obfuscated on client side then, it is transferred to the database server available at service provider's end for storage. The result of all the three scenarios is shown in the graph displayed in fig 2. Series 3 is representing the execution time of scenario 1(Plain data) , series 2 representing the execution time of scenario 3 (Encrypted plus Obfuscated data) i.e proposed model and similarly, series 3 representing the execution time of scenario 2 (Only encrypted data).

After the end of this experiment we conclude that the performance is increased by 33% with our proposed model as compared to those using only encryption technique to achieve confidentiality.

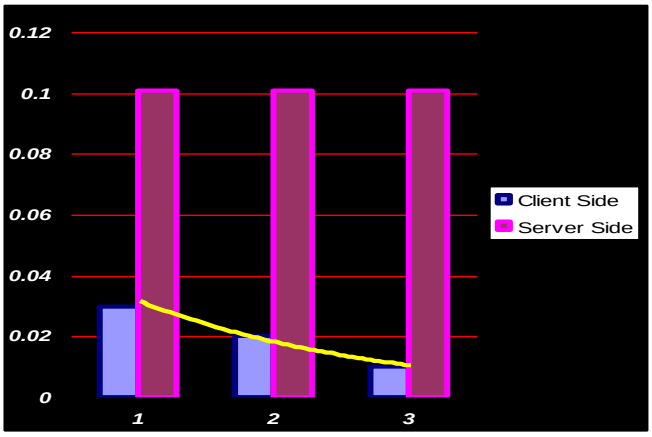

Fig 2: Record Insertion (Graph)

5.2 Experiment 2 (query performance)

Experiment 2 is performed to validate the performance enhancement during the execution of query. In this series of experiment we executed queries over plain text, over encrypted data and at the end encrypted plus obfuscated data. The result of all the three tests under this experiment is shown in graph displayed in fig 3. Series 1 is representing the time used to

execute query over plain data, as executes traditionally. Series 2 is representing the time used to execute the same query over confidential data as per our proposed approach. Series 3 is representing the time required to execute the same query over encrypted data (i.e without obfuscation feature).

After the end of this experiment we conclude that it is obvious that the time for query over confidential data will be increased as compared to the query over un-confidential data. But the time for query over encrypted data will be decreased if the element of obfuscation is added. So as per our proposed model the query performance is increased by 20% as compared to the query over encrypted data.

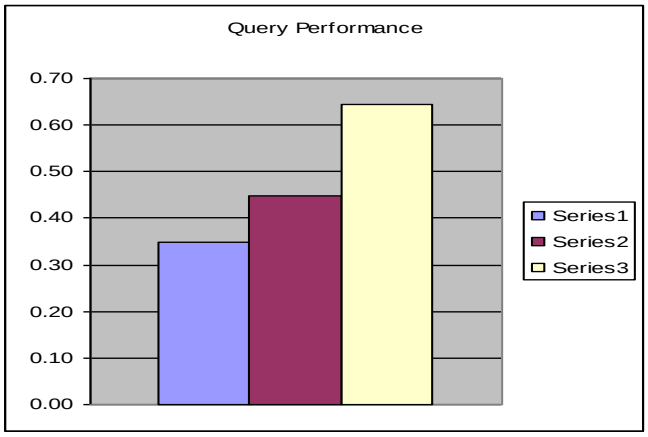

Fig 3: Comparison Graph for Query Execution

The results of experiment one and two are showing the noticeable performance enhancement due to the obfuscation element on numeric type of data.

6. Conclusion

We have presented a model to preserve confidentiality of data stored in cloud databases like DaaS. Our model has two main features. The first one covers that how to store data into the DaaS. Second feature covers that how to query data from DaaS so that confidentiality of data could not be compromised specially by the database administrators.

As far as data storage is concerned, our model has proposed to encrypt character based sensitive columns through any algorithm on client side before sending to server for storage. Similarly, it also describes to obfuscate numeric type sensitive columns through any mathematical function on client side before sending to server for storage. The mathematical function required for obfuscation should be the one by which ranking of data could be preserved, like we have used square root function for this purpose.

Moreover the proposed model focuses on to query over encrypted plus obfuscated data. All the queries are transformed on client side to execute over encrypted and obfuscated data, stored into the database server of cloud service provider. Once the required data is filtered out then it is sent to the client end where it is decrypted and de-obfuscated before presentation to the terminal. Because of obfuscation feature aggregate type of queries are easy to execute as compared to the environment where numeric type data is also encrypted to preserve confidentiality of data.

At the end we have validated the proposed approach through simulation. We have compared the proposed storage and query mechanism with traditional (non confidential) as well as the environment where confidentiality is achieved through encryption. The results of experiments conclude that the performance is decreased as compared to the un-confidential data but it is noticeably increased from the environment where only encryption technique is used to achieve confidentiality.

7. Future work

DaaS model works under the category of SaaS delivery model of cloud computing, which provides minimum control to data owner as compared to other delivery models of cloud computing. In addition to day's encryption techniques are useful especially for the environment that is not public. As cloud environment is open and it is also strongly believe that today's modern encryption algorithm are likely to be broken in near future. In future we intend to work in this direction so that if encryption algorithm is compromised, even then confidentiality could not be compromised.

References

[1] H. Hacıgumus et al, "Executing SQL over encrypted Data in the Database-Service-Provider Model" ACM SIGMOD, 2002.

[2] Miranda Mowbray and Sanani Person, "A client based privacy manager for Cloud Computing" ICST COMSWARE, 2009.

[3] Hong, Jing et al, "Executing Query over Encrypted Character Strings in Databases" National Hi-Tech Research and Development Program, 2006.

[4] Zheng-Fei et al, "Storage and query over encrypted character and numeric data in database", CIT, 2005.

[5] C.Wang, K.Ren , Q.Wang, and W.Lou, "Ensuring Data Storage Security in Cloud Computing" IEEE IWQOS, 2009.

[6] W. Wang and Z. Li, "Secure and efficient access to outsourced Data" ACM CCSW, 2009.

[7] N. Santos, R. Rodrigues and K. P. Gummadi, "Towards trusted cloud computing" un-published.

[8] D. Lin and A. squicciarini "Data Protection Models for Service provisioning in the Cloud" ACM SACMAT, 2010.

[9] W. Itani, A. Kayssi and A. Chehab, "Privacy as a service: Privacy aware data storage and processing in cloud computing architecture" IEEE DASC, 2009.

[10] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of Cloud computing" J Network Comput Appl, 2010.

Author

Mr. Atiq ur Rehman is a student of MS (Computer Science) in Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology (SZABIST) Islamabad, Pakistan. He has completed MCS (master's in computer science) from SZABIST in 2003. He also holds number of others national and international certificates in computing field. He is the author of "Cloud Security Auditing" research paper, accepted and published in 2nd International Conference of Next Generation Information Technology (ICNIT) 2011 (IEEE Korea). Presently, the author is working in a Government IT organization as database administrator.