

A Comparative Study of Attacks on Databases and Database Security Techniques

A.W Akanji

Computer Science Department
Lagos State Polytechnic
Lagos, Nigeria
akanjiwasiu2005@yahoo.com

A.A. Elusoji & A.V. Haastrup PhD

Computer Technology Department
Yaba College of Technology
Yaba, Lagos, Nigeria.
elusoji872@yahoo.com, victleye@gmail.com

ABSTRACT

Security has become one of the important challenges that people are facing all over the world in every aspect of their lives likewise security in electronic world has a great significance. Present day global business environment presents numerous security threats and compliance challenges. To protect against data thefts and frauds, we require security solutions that are transparent by design. Data is most important in today's world as it helps organizations as well as individuals to extract information and use it to make various decisions. Data are generally stored in database so that retrieving and maintaining it becomes easy and manageable. In this paper, concise review of major threats in database security, database security techniques along with their usage is presented and security policy also that should be enforced to reduce and eliminate the security threats.

Keywords — Database, Access Control, Encryption, Security

African Journal of Computing & ICT Reference Format:

A.W. Akanji, A.A. Elusoji & A.V. Haastrup (2014). A Comparative Study of Attacks on Databases and Database Security Techniques. Afr J. of Comp & ICTs. Vol 7, No. 5. Pp1-8.

I. INTRODUCTION

Data or information is the major component on which entire organization depends. It is an important asset in any organization. Almost all organization like social, governmental, educational etc, have now automated their information systems and other operational functions. They have maintained the databases which contain the crucial information. So database security is a serious concern. This dependency is so intense that success and failure of organization's goals relies on the quality and quantity of data. So naturally organizations can't afford to lose vital data present about the organization and its business. Major chunk of data are stored in the repository called database [6][17]. The data stored in databases will be structured and generally stored in the form of relational tables as most of the organizations use relational databases. As relational data model is used, data stored in different relational tables are related to each other. Protecting the confidential data stored in a repository is actually the database security.

It will secure the databases from any form of illegal access or threat at any level. Database security demands prohibiting or permitting user actions on the database and the objects inside it. Enterprises or organizations which are running successfully demand the confidentiality of their database. They do not allow the unauthorized access to their information and they also demand the surety that their data is protected against any malicious or accidental modification. As data stored in databases may be critical, it is important to secure it. Database can be attacked in many ways. There is a possibility of attacking data stored in databases as databases are interfaced with some applications and by hampering the applications; it is possible to attack databases. [3][4]. The situation becomes critical when users of database are leaking the information to outside world. Computer Security always addresses three important aspects of computer related system namely Confidentiality, Integrity and Availability. Figure 1 below shows the properties of database security that are integrity, confidentiality and availability [6][7][8].

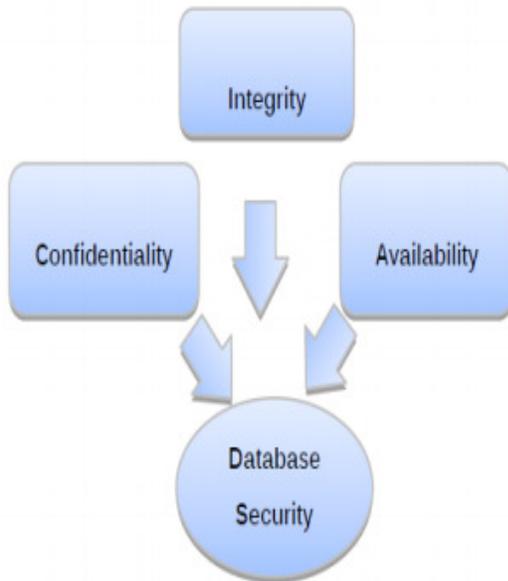


Figure 1: Properties of Database Security

Confidentiality ensures that computer related assets are accessed only by authorized users. Integrity means computer assets can be modified by authenticated users in the authorized ways. Availability ensures that assets are accessible to authorized users at appropriate times [1]. Database is a computer asset so confidentiality, integrity and availability should be considered before applying any security policy on database systems.

2. RELATED WORK ON DATABASE SECURITY TECHNIQUES.

A. Securing Database using Cryptography

Sesay et al. proposed a database encryption scheme. In this scheme the users are divided into two levels: Level 1 (L1) and Level 2 (L2). Level 1 users have access to their own private encrypted data and the unclassified public data, whereas Level 2 users have access to their own private data and also classified data which is stored in an encrypted form. Liu et al. proposed a novel database encryption mechanism [10]. The proposed mechanism performs column-wise encryption that allows the users to classify the data into sensitive data and public data. This classification helps in selecting to encrypt only that data which is critical and leaves the public data untouched thereby reducing the burden of encrypting and decrypting the whole database, as result of which the performance is not degraded. Mixed Cryptography Database [1] scheme is presented by Kadhemi et al. The technique involves designing a framework to encrypt the databases over the unsecured network in a diversified form that comprise of owning many keys by various parties. In the proposed framework, the data is grouped depending upon the ownership and on other conditions.[5].

B. Securing Database using Steganography

Das et al. explained various techniques in steganography that can be implemented to hide critical data and prevent them from unauthorized and direct access. The various techniques include still image steganography, audio steganography, video steganography, IP Datagram steganography. Naseem et al. presented a method that uses steganography to hide data. In the proposed scheme the data is embedded in the LSB's of the pixel values.

The pixels values are categorized into different ranges and depending on the range certain number of bits is allocated to hide the sensitive data. Kuo et al. presented a different approach to conceal data. In this scheme the image is divided into fixed number of blocks. Histogram of each block is calculated along with the maximum and minimum points to mask the data. This mechanism increases the hiding capacity of the data.[9]. Dey et al. employs a diverse approach to efficiently hide the sensitive data and escalate the data hiding capacity in still images. The technique involves using prime numbers and natural numbers to enhance the number of bit planes to cloak the data in the images.

C. Securing Database using Access Control

Bertino et al. explains an authorization technique for video databases. In the proposed scheme, the access to the database and to a particular stream of the video is granted only after verifying the credentials of that user. The credentials may not just be the user-id but it may be the characteristics that define the user and only after successful verification of the credentials the user is granted the permission to access the database.

Kodali et al. presented a generalized authorization model for multimedia digital libraries. The scheme involves integrating the three most common and widely used access control mechanisms namely: mandatory, discretionary and role-based models into a single framework to allow a unified access to the protected data. The technique also addresses the need of continuous media data while supporting the QoS constraints alongside preserving the operational semantics. An authorization model is proposed by Rizvi et al. In the explained technique is based on authorization views which enable authorization transparent querying in which the user queries are formed and represented in terms of database relations and are acceptable only when the queries can be verified using the information contained in the authorization rules. The work presents the new techniques of validity and conditional validity which is an extension of the earlier work done in the same area.

3. SECURITY THREATS IN DATABASE

1. *Excessive and Unused Privileges*

When someone is granted database privileges that exceed the requirements of their job function, these privileges can be abused. For example, a bank employee whose job requires the ability to change only account holder contact information may take advantage of excessive database privileges and increase the account balance of a colleague's savings account. Further, when someone leaves an organization, often his or her access rights to sensitive data do not change. And, if these workers depart on bad terms, they can use their old privileges to steal high value data or inflict damage. Users end up with excessive privileges because privilege control mechanisms for job roles have not been well defined or maintained. As a result, users may be granted generic or default access privileges that far exceed their specific job requirements. This creates unnecessary risk.

2. *Privilege Abuse*

Users will abuse legitimate database privileges for unauthorized purposes. Consider an internal healthcare application used to view individual patient records via a custom Web interface. The Web application normally limits users to viewing an individual patient's healthcare history – multiple patient records cannot be viewed simultaneously and electronic copies are not allowed. However, a rogue user might be able to circumvent these restrictions by connecting to the database using an alternative client such as MS-Excel. Using Excel and their legitimate login credentials, the user could retrieve and save all patient records to their laptop.[13] Once patient records reach a client machine, the data then becomes susceptible to a wide variety of possible breach scenarios.

3. *Input Injection (Formerly SQL Injection)*

There are two major types of database injection attacks: 1) SQL Injection that targets traditional database systems and 2) NoSQL Injection that targets Big Data platforms. SQL Injection attacks usually involve inserting (or “injecting”) unauthorized or malicious statements into the input fields of Web applications. On the other hand, NoSQL injection attacks involve inserting malicious statements into Big Data components (e.g., Hive, MapReduce, etc.). A successful Input Injection attack can give an attacker unrestricted access to an entire database.[11][12].

It is important to note that there are misconceptions about Big Data being impervious to SQL Injection attacks. These misconceptions are partly true due to the fact that Big Data does not leverage SQL-based technologies. However, as mentioned earlier, Big Data's underlying components are still susceptible to Input Injection attacks.

4. *Malware*

Cybercriminals, state-sponsored hackers, and spies use advanced attacks that blend multiple tactics – such as spear phishing emails and malware – to penetrate organizations and steal sensitive data. Unaware that malware has infected their device, legitimate users become a conduit for these groups to access your networks and sensitive data.

5. *Weak Audit Trail*

Automated recording of database transactions involving sensitive data should be part of any database deployment. Failure to collect detailed audit records of database activity represents a serious organizational risk on many levels. Organizations with weak (or sometimes non-existent) database audit mechanisms will increasingly find that they are at odds with industry and government regulatory requirements.[16] For example, Sarbanes-Oxley (SOX), which protects against accounting errors and fraudulent practices, and the Healthcare Information Portability and Accountability Act (HIPAA) in the healthcare sector, are just two examples of regulations with clear database audit requirements.

Many enterprises will turn to native audit tools provided by their database vendors or rely on ad-hoc and manual solutions. These approaches do not record details necessary to support auditing, attack detection, and forensics. Furthermore, native database audit mechanisms are notorious for consuming CPU and disk resources forcing many organizations to scale back or eliminate auditing altogether. Finally, most native audit mechanisms are unique to a database server platform. For example, Oracle logs are different from MS-SQL, and MS-SQL logs are different from DB2. For organizations with heterogeneous database environments, this imposes a significant obstacle to implementing uniform, scalable audit processes.

When users access the database via enterprise Web applications (such as SAP, Oracle E-Business Suite, or PeopleSoft) it can be challenging to understand what database access activity relates to a specific user. Most audit mechanisms have no awareness of who the end user is because all activity is associated with the Web application account name. Reporting, visibility, and forensic analysis are hampered because there is no link to the responsible user.[14] Finally, users with administrative access to the database, either legitimately or maliciously obtained, can turn off native database auditing to hide fraudulent activity. Audit duties should ideally be separate from both database administrators and the database server platform to ensure strong separation of duties policies.

6. Storage Media Exposure

Backup storage media is often completely unprotected from attack. As a result, numerous security breaches have involved the theft of database backup disks and tapes. Furthermore, failure to audit and monitor the activities of administrators who have low-level access to sensitive information can put your data at risk. Taking the appropriate measures to protect backup copies of sensitive data and monitor your most highly privileged users is not only a data security best practice, but also mandated by many regulations.

7. Exploitation of Vulnerable, Mis-configured Databases

It is common to find vulnerable and un-patched databases, or discover databases that still have default accounts and configuration parameters. Attackers know how to exploit these vulnerabilities to launch attacks against your organization. Unfortunately, organizations often struggle to stay on-top of maintaining database configurations even when patches are available. It generally takes organizations months to patch databases once a patch is available. During the time your databases are un-patched, they remain vulnerable. According to the 2012 Independent Oracle User Group (IOUG), 28 percent of Oracle users have never applied a Critical Patch Update or don't know whether they've done so. Another 10 percent take a year or longer to apply their patches [15].

8. Unmanaged Sensitive Data

Many companies struggle to maintain an accurate inventory of their databases and the critical data objects contained within them. Forgotten databases may contain sensitive information, and new databases can emerge – e.g., in application testing environments – without visibility to the security team. Sensitive data in these databases will be exposed to threats if the required controls and permissions are not implemented.

9. Denial of Service

Denial of Service (DoS) is a general attack category in which access to network applications or data is denied to intended users. DoS conditions can be created via many techniques. The most common technique used in database environments is to overload server resources such as memory and CPU by flooding the network with database queries that ultimately cause the server to crash. The motivations behind DoS attacks are often linked to extortion scams in which a remote attacker will repeatedly crash servers until the victim meets their demands. Whatever the source, DoS represents a serious threat for many organizations.

10. Limited Security Expertise and Education

Internal security controls are not keeping pace with data growth and many organizations are ill-equipped to deal with a security breach. Often this is due to the lack of expertise required to implement security controls, policies, and training. According to PWC's 2012 Information Security Breaches Survey, 75% of the organizations surveyed experienced staff-related breaches when a security policy was poorly understood and 54% of small businesses did not have a program for educating their staff about security risks.

4. DATABASE SECURITY CONSIDERATIONS

To eliminate the security threats every organization must define a security policy also that should be strictly enforced. A strong security policy must contain well defined security features. Figure 2 shows some critical areas that need to be considered are explained below [1][3][4].

a. Access Control

Access control ensures that all communication with the databases and other system objects are according to the policies and controls defined. This makes sure that no interference occurs by any attacker neither internally nor externally and thus, protects the databases from potential errors that can make impact as big as stopping firms operations. Access control also helps in minimizing the risks that may directly impact the security of the database on the main servers. For example, if any table is accidentally deleted or access is modified the results can be roll backed or for certain files access control can restrict their deletion.

b. Inference Policy

It is required to protect the data at a certain level. It occurs when the interpretations from certain data in the form of analysis or facts are required to be protected at a higher security level. It also determines how to protect the information from being disclosed.

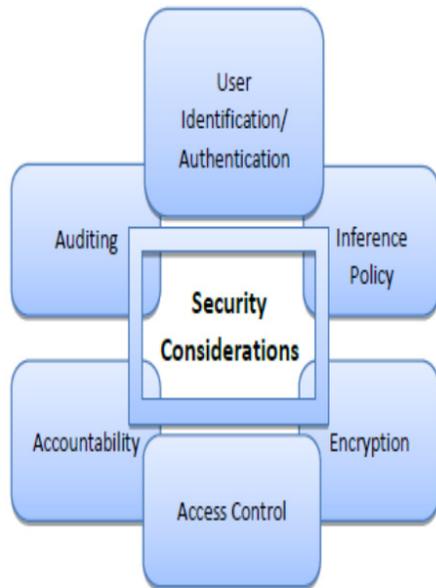


Fig 2: Critical areas under consideration

c. User Identification Authentication

User identification and authentication is the basic necessity to ensure security since the identification method defines a set of people that are allowed to access data and provides a complete mechanism of accessibility. To ensure security, the identity is authenticated and it keeps the sensitive data safe and from being modified by any ordinary user.

d. Accountability and Auditing

Accountability and audit checks are required to ensure physical integrity of the data which requires defined access to the databases and that is managed through auditing and record keeping. It also helps in analysis of information held on servers for authentication, accounting and access of a user.[15].

e. Encryption

This is the basic technique used for securing any kind of information or data. So this technique can even be applied to databases.



Figure 3: Basic encryption processes

Encryption is a process of translating plain text to encoded form called cipher text or a code so that it becomes unreadable to all other people except those who hold a key to the information. The resulting encoded information is called encrypted information. This is usually carried out using secret encryption key and cryptographic cipher. Figure 3 illustrates the basic process of encryption. Data are encrypted using encryption keys and encryption algorithms. Encrypted data are then stored in the database and decrypted when need to be used for processing purpose.[16].

There are two basic types of encryption commonly used. Symmetric Encryption is the type of encryption where a single secret key is used for both encryption and decryption. Asymmetric encryption is the type of encryption where a pair of secret keys is used. One of the keys is used for encryption and the other used for decryption. While performing database encryption, a decision about whether to perform the encryption inside or outside the database must be taken. Some of the issues involved in this technique are How to secure keys from attacker of the system? How to give administrative rights of manipulating data using keys? And How to provide limited access for keys?

It is also important to provide proper authentication mechanisms because without them, it is easy to get access to keys using social engineering techniques [7]. [6] Though encryption improves the protection but its implementation decisions are also very important. Following figure 4 shows where encryption takes place. Developing the encryption strategies arises some important questions also, like how, when and where the encryption will be performed.

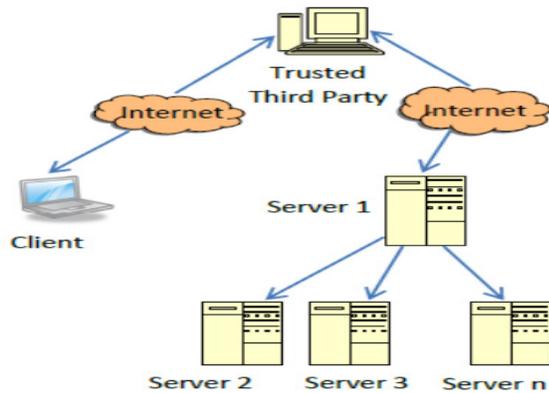


Fig 4: Three levels where encryption is performed

The important aspects which need to be considered while encrypting database is how to manage the encryption keys. Some of the aspects related to this issue are Number of encryption keys required, storage of keys, protection for the access of keys, and frequency of change of keys. Recommended approach for storing the keys is, separate the keys and data residing in the database. Generally the keys are stored in hardware like access restricted files or hardware storage modules.[18][19] The process of encryption can be performed either within the database or outside the database. If encryption is performed within the database, then there is less impact on application environment. But there are performance and security tradeoffs which need to be considered while implementing this policy. Understanding the encryption algorithm supported by DBMS also plays key role while devising strategy to implement this technique. The drawback of this approach is encryption keys also are stored in the same database.

Another way to implement encryption in database is performing it on separate encryption servers. Encryption and decryption computations are performed encryption server. So here overhead of encryption is removed from DBMS and moved on to separate encryption servers to maintain the performance of DBMS. Encryption keys and data can also be separated. This approach is usually followed while encrypting database [7]. The algorithms which are generally used for database encryption and often supported by DBMS are DES, Triple DES, RC2, RC4, DESX and AES.

The database encryption scheme can be implemented using different approaches. There are two main things to consider while considering database encryption. First thing is granularity of the data to be encrypted or decrypted. [11] Granularity can be field level, row level or page level. Row or page level granularity may lead to encrypting large amount of data which can be overhead on the system. So generally column level encryption of only sensitive data is performed. The second thing is choice of encryption algorithm which is suitable for encrypting given data in database [8].

One encryption system approach describes two phases called initialization phase and run phase. In the initialization phase, all the metadata like the columns to be encrypted, the type and length of the columns, encryption algorithm and encrypted columns on which index is required. Such metadata is stored in the Security Dictionary. It will be loaded into memory first time it is used.[19]. In the run phase of this scheme, the application does the normal activities performed on the database without thinking about encryption. Encryption/decryption engine performs data encryption and decryption based on metadata stored in Security Dictionary [8]. There are various configurations available for encrypting and decrypting databases. Some of them are listed below :-

File System Encryption: Here the physical disk where database resides is encrypted. Entire database is encrypted using single encryption key so discretionary access control cannot be implemented.

DBMS Level Encryption: There are many schemes for this kind of encryption. One scheme is based on Chinese Remainder theorem in which every row is encrypted using different sub keys for different cells. So encryption at row level and decryption at cell or field level is possible by this scheme.

There are some schemes based on Newton's interpolation polynomials which are used for database encryption. [21]. There is a SPDE scheme which encrypts each cell I the database with its cell coordinates like table name, column name and row id etc. So in this scheme static leakage attacks and splicing attacks are prevented.

Application level Encryption: In this technique, a middleware is suggested which translates queries fired by user into new bunch of queries which will execute on encrypted database. This technique was implemented in Data Protector System.

Client-side encryption: This technique is generally used in case of —Database as a service scenario where the entire database is outsourced by the organization to reduce the maintenance costs. So here data privacy is the major concern. Encryption is the basic solution in this scenario. **Indexing encrypted data:** There are many indexing mechanisms proposed. B tree index structure is prepared over plain text values in the table and then encryption of the table is performed at the row level. Encryption of the Btree is done at the node level.[20].

Another scheme involves constructing index on plain text values and then encryption of each page of the index is done separately. One more modification is suggested which involves encrypting different index pages with different keys depending on page number. There is another scheme suggested which computes XOR of plain text values with sequence of pseudo random bits which are generated by the client according to plain text value and a secure encryption keys.

A database encryption system must adhere to some characteristics such as it should be secure enough so that it requires high work factor to break, encryption and decryption should be performed fast without compromising DBMS performance, encrypted data should be small compared to unencrypted data, it should be possible to perform encryption and decryption of records without taking into consideration their physical or logical position in database, encryption scheme must support logical sub schema concepts of databases, encrypted record should be one value which is function of all fields, the encryption scheme should be as flexible as possible with respect to combinations of read and write operations, encryption system should not force DBMS to keep duplicate copies of data so that sub schema should be supported [9].

5. CONCLUSION

Databases form the backbone of many applications today. Data to any organization is most valuable property. Security of sensitive data is always a big challenge for an organization at any level. They are the primary form of storage for many organizations. In today's technological world, database is vulnerable to hosts of attacks hence the attacks on databases are also increasing as they are very dangerous form of attack. They reveal key or important data to the attacker. Various attacks on databases are discussed in this paper. This research will lead to more concrete solution for database security issue

REFERENCES

- [1] Kadhem, H.; Amagasa, T.; Kitagawa, H.; A Novel Framework for Database Security based on Mixed Cryptography; Internet and Web Applications and Services, 2009. ICIW '09. Fourth International Conference on; Publication Year: 2009, Page(s): 163 –170
- [2] Luc Bouganim; Yanli GUO; Database Encryption; Encyclopedia of Cryptography and Security, S. Jajodia and H. van Tilborg (Ed.) 2009, page(s):) 1-9
- [3] Khaleel Ahmad; Jayant Shekhar; Nitesh Kumar; K.P. Yadav; Policy Levels Concerning Database Security;
- [4] International Journal of Computer Science & Emerging Technologies (E-ISSN: 2044-6004) 368 Volume 2, Issue 3, June 2011, page(s); 368-372.
- [5] Iqra Basharat, Farooque Azam, Abdul Wahab Muzaffar, "Database Security and Encryption: A Survey Study", International Journal of Computer Applications (0975 – 888) Volume 47– No.12, June 2012.
- [6] Mr. Saurabh Kulkarni, Dr. Siddhaling Urolagin, "Review of Attacks on Databases and Database Security Techniques", International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, Volume 2, Issue 11, November 2012.
- [7] Emil Burtescu, "DATABASE SECURITY - ATTACKS AND CONTROL METHODS", Journal of Applied Quantitative Methods, Vol. 4, no. 4, Winter 2009.
- [8] Ahmad Baraani-Dastjerdi; Josef Pieprzyk; Baraanidastjerdi Josef Pieprzyk ; Reihaned Safavi-Naini, Security In Databases: A Survey Study, 1996
- [9] Amichai Shulman; Top Ten Database Security Threats, How to Mitigate the Most Significant Database Vulnerabilities, 2006 White Paper.
- [10] Tanya Bacca; Making Database Security an IT Security Priority A SANS Whitepaper – November 2009
- [11] E. Anupriya, Sachin Soni, Amit Agnihotri, Sourabh Babelay, "Encryption using XOR based Extended Key for Information Security – A Novel Approach", International Journal on Computer Science and Engineering (IJCSE), vol. 3, issue 1, Jan. 2011, pp. 146-154
- [12] Ahmad Baraani-Dastjerdi; Josef Pieprzyk; Baraani- dastjerdi Josef Pieprzyk ; Reihaned Safavi-Naini, Security In Databases: A Survey Study, 1996
- [13] Amichai Shulman; Top Ten Database Security Threats, How to Mitigate the Most Significant Database Vulnerabilities, 2006 White Paper.
- [14] Tanya Bacca; Making Database Security an IT Security Priority A SANS Whitepaper – November 2009

- [15] Kadhem, H.; Amagasa, T.; Kitagawa, H.; A Novel Framework for Database Security based on Mixed Conference on; Publication Year: 2009, Page(s): 163- 170
- [16] Luc Bouganim; Yanli GUO; Database Encryption; Encyclopedia of Cryptography and Security, S. Jajodia and H. van Tilborg (Ed.) 2009, page(s):) 1-9
- [17] Khaleel Ahmad; JayantShekhar; Nitesh Kumar; K.P. Yadav; Policy Levels Concerning Database Security; International Journal of Computer Science & Emerging Technologies (E-ISSN: 2044-6004) 368 Volume 2, Issue 3, June 2011, page(s); 368-372
- [18] Gang Chen; Ke Chen; Jinxiang Dong; A Database Encryption Scheme for Enhanced Security and Easy Sharing; Computer Supported Cooperative Work in Design, 2006. CSCWD '06. 10th International Conference on ; Publishing year 2006, page(s): 1 – 6
- [19] Dr. Anwar Pasha Abdul GafoorDeshmukh; Dr. Anwar Pasha Abdul afoorDeshmukh; Transparent Data Encryption- Solution for Security of Database Contents; (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No.3, March 2011
- [20] TingjianGe, Stan Zdonik; Fast, Secure Encryption for Indexing in a Column-Oriented DBMS; 2007 IEEE 23rd International Conference on Data Engineering (2007) Publisher: IEEE, Page(s): 676-685.
- [21] Lianzhong Liu and JingfenGai; A New Lightweight Database Encryption Scheme Transparent to Applications; Published in Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference Issue Date: 13-16 July 2008 On page(s): 135 – 140