International Journal On Engineering Technology and Sciences – IJETS™
ISSN (P): 2349-3968, ISSN (O): 2349-3976
Volume 1 Issue 6, October 2014

Synchronized Encrypted Database Structure Management Method for Remote Storage Environment

Ms. S.Tharani
II Year M.E(CSE)
Shree Venkateshwara Hi-Tech Engg College, Gobi
tharanimahalakshmi@gmail.com

Dr. T. Senthil Prakash
Professor & HOD
Shree Venkateshwara Hi-Tech Engg College, Gobi
jtyesp@yahoo.co.in

Mr. S. Prakadeswaran, M.E.,
Assistant Professor
Shree Venkateshwara Hi-Tech Engg College, Gobi
prakades@gmail.com

Abstract - *Cloud computing enables highly scalable services to consumed over the Internet. Cloud services are provided on user request basis. DataBase as a Service (DBaaS) model is used to manage databases in cloud environment. Secure DBaaS model provides data confidentiality for cloud databases. Secure DBaaS is designed to allow multiple and independent clients to connect to the cloud without intermediate server. Data, data structures and metadata are encrypted before upload to the cloud. Multiple cryptography techniques are used to convert plain text into encrypted data. Table names and their column names are also encrypted in the cloud database security scheme. The system supports geographically distributed clients to connect directly to an encrypted cloud database. The client can perform concurrent query processing on encrypted databases. RSA and Advanced Encryption Standard (AES) are used in the system. The Secure DBaaS framework is enhanced to support concurrent database structure modification scheme with minimum overhead. Digital signature based data integrity verification mechanism is integrated with the system. Encrypted query submission model is used to secure the query values. Access control mechanism is used to allow users to grant permissions for other users.*

Index Terms – *Secure DBaaS, Database structure, cloud database, AES, Encrypted query*

1 INTRODUCTION

Cloud computing is a recent trend in IT that moves computing and data away from desktop and portable PCs into large data centers. It refers to applications delivered as services over the Internet as well as to the actual cloud infrastructure — namely, the hardware and systems software in data centers that provide these services. The key driving forces behind cloud computing are the ubiquity of broadband and wireless networking, falling storage costs, and progressive improvements in Internet computing software. Cloud-service clients will be able to add more capacity at peak demand, reduce costs, experiment with new services and remove unneeded capacity, whereas service providers will increase utilization via multiplexing, and allow for larger investments in software and hardware.

Currently, the main technical underpinnings of cloud computing infrastructures and services include virtualization, service-oriented software, grid computing technologies, management of large facilities, and power efficiency. Consumers purchase such services in the form of infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), or soft-ware-as-a-service (SaaS) and sell value-added services to users. Within the cloud, the laws of probability give service providers great leverage through statistical multiplexing of varying workloads and easier management - a single software installation can cover many users' needs.

We can distinguish two different architectural models for clouds: the first one is designed to scale out by providing additional computing instances on demand. Clouds can use these instances to supply services in the form of SaaS and PaaS. The second architectural model is designed to provide data and compute-intensive applications via scaling capacity. In most cases, clouds provide on-demand computing instances or capacities with a "pay-as-you-go" economic model. The cloud infrastructure can support any computing model compatible with loosely coupled CPU clusters. Organizations can provide hardware for clouds internally, or a third party can provide it externally. A cloud might be restricted to a single organization or group, available to the general public over the Internet, or shared by multiple groups or organizations.

A cloud comprises processing, network, and storage elements, and cloud architecture consists of three abstract layers. Infrastructure is the lowest layer and is a means of delivering basic storage and compute capabilities as standardized services over the network. Servers, storage systems, switches, routers, and other systems handle specific types of workloads, from batch processing to server or storage

augmentation during peak loads. The middle platform layer provides higher abstractions and services to develop, test, deploy, host, and maintain applications in the same integrated development environment. The application layer is the highest layer and features a complete application offered as a service.

2 RELATED WORK

Secure DBaaS provides several original features that differentiate it from previous work in the field of security for remote database services.

- It guarantees data confidentiality by allowing a cloud database server to execute concurrent SQL operations over encrypted data.
- It provides the same availability, elasticity, and scalability of the original cloud DBaaS because it does not require any intermediate server. Response times are affected by cryptographic overheads that for most SQL operations are masked by network latencies.
- Multiple clients, possibly geographically distributed, can access concurrently and independently a cloud database service.
- It does not require a trusted broker or a trusted proxy because tenant data and metadata stored by the cloud database are always encrypted.
- It is compatible with the most popular relational database servers, and it is applicable to different DBMS implementations because all adopted solutions are database agnostic.

Cryptographic file systems and secure storage solutions represent the earliest works in this field. We do not detail the several papers and products (e.g., Sporc [3], Sundr, Depot [5]) because they do not support computations on encrypted data.

Different approaches guarantee some confidentiality (e.g., [4]) by distributing data among different providers and by taking advantage of secret sharing. In such a way, they prevent one cloud provider to read its portion of data, but information can be reconstructed by colluding cloud providers. A step forward is proposed in [7], that makes it possible to execute range queries on data and to be robust against collusive providers. SecureDBaaS differs from these solutions as it does not require the use of multiple cloud providers, and makes use of SQL-aware encryption algorithms to support the execution of most common SQL operations on encrypted data.

SecureDBaaS relates more closely to works using encryption to protect data managed by untrusted databases. In such a case, a main issue to address is that cryptographic techniques cannot be naıvely applied to standard DBaaS because DBMS can only execute SQL operations over plaintext data. Some DBMS engines offer the possibility of encrypting data at the filesystem level through the so-called Transparent Data Encryption feature [6]. This feature makes it possible to build a trusted DBMS over untrusted storage. The DBMS is trusted and decrypts data before their use. Hence, this approach is not applicable to the DBaaS context considered by SecureDBaas, because we assume that the cloud provider is untrusted.

Other solutions, allow the execution of operations over encrypted data. These approaches preserve data confidentiality in scenarios where the DBMS is not trusted; however, they require a modified DBMS engine and are not compatible with DBMS software (both commercial and open source) used by cloud providers. On the other hand, SecureDBaaS is compatible with standard DBMS engines, and allows tenants to build secure cloud databases by leveraging cloud DBaaS services already available. SecureDBaaS is more related to [8] that preserve data confidentiality in untrusted DBMSs through encryption techniques, allow the execution of SQL operations over encrypted data, and are compatible with common DBMS engines. The architecture of these solutions is based on an intermediate and trusted proxy that mediates any interaction between each client and the untrusted DBMS server. The approach proposed the authors of the DBaaS model works by encrypting blocks of data instead of each data item. Whenever a data item that belongs to a block is required, the trusted proxy needs to retrieve the whole block, to decrypt it, and to filter out unnecessary data that belong to the same block. As a consequence, this design choice requires heavy modifications of the original SQL operations produced by each client, thus causing significant overheads on both the DBMS server and the trusted proxy. Other works introduce optimization and generalization that extend the subset of SQL operators supported but they share the same proxy-based architecture and its intrinsic issues. On the other hand, SecureDBaaS allows the execution of operations over encrypted data through SQL-aware encryption algorithms. This technique, initially proposed in CryptDB makes it possible to execute operations over encrypted data that are similar to operations over plaintext data. In many cases, the query plan executed by the DBMS for encrypted and plaintext data is the same.

The reliance on a trusted proxy that characterize facilitates the implementation of a secure DBaaS, and is applicable to multitier web applications, which are their main focus. It causes several drawbacks. Since the proxy is trusted, its functions cannot be outsourced to an untrusted cloud provider. The proxy is meant to be implemented and managed by the cloud tenant. Availability, scalability and elasticity of

the whole secure DBaaS service are then bounded by availability, scalability and elasticity of the trusted proxy, that becomes a single point of failure and a system bottleneck. Since high availability, scalability and elasticity are among the foremost reasons that lead to the adoption of cloud services, this limitation hinders the applicability to the cloud database scenario. SecureDBaaS solves this problem by letting clients connect directly to the cloud DBaaS, without the need of any intermediate component and without introducing new bottlenecks and single points of failure.

A proxy-based architecture requiring that any client operation should pass through one intermediate server is not suitable to cloud-based scenarios, in which multiple clients, typically distributed among different locations, need concurrent access to data stored in the same DBMS. On the other hand, SecureDBaaS supports distributed clients issuing independent and concurrent SQL operations to the same database and possibly to the same data. SecureDBaaS extends our preliminary studies [9] showing that data consistency can be guaranteed for some operations by leveraging concurrency isolation mechanisms implemented in DBMS engines, and identifying the minimum isolation level required for those statements. Moreover, we now consider theoretically and experimentally a complete set of SQL operations represented by the TPC-C standard benchmark [10], in addition to multiple clients and different client-cloud network latencies that were never evaluated in the literature.

3 DISTRIBUTED AND CONCURRENT ACCESS TO ENCRYPTED CLOUD DATABASES

In a cloud context, where critical information is placed in infrastructures of untrusted third parties, ensuring data confidentiality is of paramount importance [1], [2]. This requirement imposes clear data management choices: original plain data must be accessible only by trusted parties that do not include cloud providers, intermediaries, and Internet; in any untrusted context, data must be encrypted. Satisfying these goals has different levels of complexity depending on the type of cloud service. There are several solutions ensuring confidentiality for the storage as a service paradigm while guaranteeing confidentiality in the database as a service (DBaaS) paradigm is still an open research area. In this context, we propose SecureDBaaS as the first solution that allows cloud tenants to take full advantage of DBaaS qualities, such as availability, reliability, and elastic scalability, without exposing unencrypted data to the cloud provider.

The architecture design was motivated by a threefold goal: to allow multiple, independent, and geographically distributed clients to execute concurrent operations on encrypted data, including SQL statements that modify the database structure; to preserve data confidentiality and consistency at the client and cloud level; to eliminate any intermediate server between the cloud client and the cloud provider. The possibility of combining availability, elasticity, and scalability of a typical cloud DBaaS with data confidentiality is demonstrated through a prototype of Secure DBaaS that supports the execution of concurrent and independent operations to the remote encrypted database from many geographically distributed clients as in any unencrypted DBaaS setup. To achieve these goals, Secure DBaaS integrates existing cryptographic schemes, isolation mechanisms, and novel strategies for management of encrypted metadata on the untrusted cloud database. This paper contains a theoretical discussion about solutions for data consistency issues due to concurrent and independent client accesses to encrypted data. In this context, we cannot apply fully homomorphic encryption schemes because of their excessive computational complexity.

The SecureDBaaS architecture is tailored to cloud platforms and does not introduce any intermediary proxy or broker server between the client and the cloud provider. Eliminating any trusted intermediate server allows SecureDBaaS to achieve the same availability, reliability, and elasticity levels of a cloud DBaaS. Other proposals based on intermediate server(s) were considered impracticable for a cloud-based solution because any proxy represents a single point of failure and a system bottleneck that limits the main benefits of a database service deployed on a cloud platform. Unlike SecureDBaaS, architectures relying on a trusted intermediate proxy do not support the most typical cloud scenario where geographically dispersed clients can concurrently issue read/write operations and data structure modifications to a cloud database.

A large set of experiments based on real cloud platforms demonstrate that SecureDBaaS is immediately applicable to any DBMS because it requires no modification to the cloud database services. Other studies where the proposed architecture is subject to the TPC-C standard benchmark for different numbers of clients and network latencies show that the performance of concurrent read and write operations not modifying the SecureDBaaS database structure is comparable to that of unencrypted cloud database. Workloads including modifications to the database structure are also supported by SecureDBaaS, but at the price of overheads that seem acceptable to achieve the desired

level of data confidentiality. The motivation of these results is that network latencies, which are typical of cloud scenarios, tend to mask the performance costs of data encryption on response time. The overall conclusions of this paper are important because for the first time they demonstrate the applicability of encryption to cloud database services in terms of feasibility and performance.

4 PROBLEM STATEMENT

Secure DBaaS is designed to allow multiple and independent clients to connect to the cloud without intermediate server. Data, data structures and metadata are encrypted before upload to the cloud. Multiple cryptography techniques are used to convert plain text into encrypted data. Table names and their column names are also encrypted in the cloud database security scheme. The system supports geographically distributed clients to connect directly to an encrypted cloud database. The client can perform concurrent query processing on encrypted databases. RSA and Advanced Encryption Standard (AES) are used in the system. Concurrent data and data structure modification are allowed on the cloud databases. The following problems are identified from the existing system.

- Database structure modification increases the computational overhead
- Data integrity verification is not performed
- Access control mechanism is not provided
- Query submission is not secured

5 SECURE DBAAS

Secure DBaaS is designed to allow multiple and independent clients to connect directly to the untrusted cloud DBaaS without any intermediate server. We assume that a tenant organization acquires a cloud database service from an untrusted DBaaS provider. The tenant then deploys one or more machines and installs a Secure DBaaS client on each of them. This client allows a user to connect to the cloud DBaaS to administer it, to read and write data, and even to create and modify the database tables after creation. We assume the same security model that is commonly adopted where tenant users are trusted, the network is untrusted and the cloud provider is honest-but-curious, cloud service operations are executed correctly, but tenant information confidentiality risk. The information managed by SecureDBaaS includes plaintext data, encrypted data, metadata, and encrypted metadata. Plaintext data consist of information that a tenant wants to store and process remotely in the cloud DBaaS. To prevent an untrusted cloud provider from violating confidentiality of tenant data stored in plain form, SecureDBaaS adopts multiple cryptographic techniques to transform plaintext data into encrypted tenant data and encrypted tenant data structures because even the names of the tables and of their columns must be encrypted. SecureDBaaS clients produce also a set of metadata consisting of information required to encrypt decrypt data as well as other administration information. Even metadata are encrypted and stored in the cloud DBaaS.

SecureDBaaS moves away from existing architectures that store just tenant data in the cloud database, and save metadata in the client machine or split metadata between the cloud database and a trusted proxy. When considering scenarios where multiple clients can access the same database concurrently previous solutions are quite inefficient. Solutions based on a trusted proxy are more feasible, but they introduce a system bottleneck that reduces availability, elasticity and scalability of cloud database services. SecureDBaaS proposes a different approach where all data and metadata are stored in the cloud database. SecureDBaaS clients can retrieve the necessary metadata from the untrusted database through SQL statements, so that multiple instances of the SecureDBaaS client can access to the untrusted cloud database independently with the guarantee of the same availability and scalability properties of typical cloud DBaaS. Encryption strategies for tenant data and innovative solutions for metadata management and storage are described.

5.1 Sequential SQL Operations

We describe the SQL operations in SecureDBaaS by considering an initial simple scenario in which we assume that the cloud database is accessed by one client. Our goal here is to highlight the main processing steps; we do not take into account performance optimizations and concurrency. The first connection of the client with the cloud DBaaS is for authentication purposes. SecureDBaaS relies on standard authentication and authorization mechanisms provided by the original DBMS server. After the authentication, a user interacts with the cloud database through the SecureDBaaS client. SecureDBaaS analyzes the original operation to identify which tables are involved and to retrieve their metadata from the cloud database. The metadata are decrypted through the master key and their information is used to translate the original plain SQL into a query that operates on the encrypted database.

Translated operations are then executed by the cloud database over the encrypted tenant data. As there is a one-to-one correspondence between plaintext tables and encrypted tables, it is possible to

prevent a trusted database user from accessing or modifying some tenant data by granting limited privileges on some tables. User privileges can be managed directly by the untrusted and encrypted cloud database. The results of the translated query that includes encrypted tenant data and metadata are received by the SecureDBaaS client, decrypted and delivered to the user. The complexity of the translation process depends on the type of SQL statement.

5.2 Concurrent SQL Operations

The support to concurrent execution of SQL statements issued by multiple independent clients is one of the most important benefits of SecureDBaaS with respect to state-of-the-art solutions. Our architecture must guarantee consistency among encrypted tenant data and encrypted metadata because corrupted or out-of-date metadata would prevent clients from decoding encrypted tenant data resulting in permanent data losses. A thorough analysis of the possible issues and solutions related to concurrent SQL operations on encrypted tenant data available in the online supplemental material. Here, we remark the importance of distinguishing two classes of statements that are supported by SecureDBaaS: SQL operations not causing modifications to the database structure, such as read, write, and update; operations involving alterations of the database structure through creation, removal and modification of database tables.

In scenarios characterized by a static database structure, SecureDBaaS allows clients to issue concurrent SQL commands to the encrypted cloud database without introducing any new consistency issues with respect to unencrypted databases. After metadata retrieval, a plaintext SQL command is translated into one SQL command operating on encrypted tenant data. As metadata do not change, a client can read them once and cache them for further uses, thus improving performance. SecureDBaaS is the first architecture that allows concurrent and consistent accesses even when there are operations that can modify the database structure. In such cases, we have to guarantee the consistency of data and metadata through isolation levels, such as the snapshot isolation that we demonstrate can work for most usage scenarios.

6 SYNCHRONIZED ENCRYPTED DATABASE MANAGEMENT METHOD FOR CLOUDS

The Secure DBaaS framework is enhanced to support concurrent database structure modification scheme with minimum overhead. Digital signature based data integrity verification mechanism is integrated with the system. Encrypted query submission model is used to secure the query values. Access control mechanism is used to allow users to grant permissions for other users. Cloud database security scheme is enhanced with data verification mechanism. Access privilege management scheme is integrated with the system. Encrypted query processing is used to secure the user query values. The system is divided into six major modules. They are cloud database, key management, data upload process, privilege management, query processing and database reconstruction.

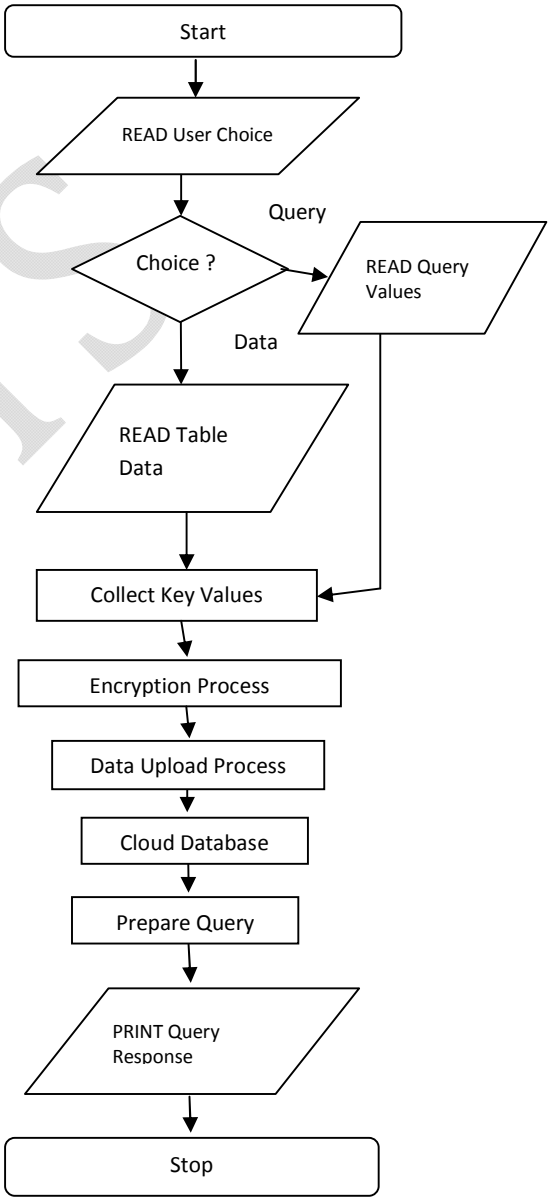

Fig. 1. Cloud Database Security Process

The cloud database module is designed to manage client data values. Key distribution process is handled under key management module. Table creation and update operations are carried out under the data upload process. User access permissions are assigned under the privilege management module. Query processing module is designed to execute encrypted queries in the cloud database. Database structure modifications are performed under the database reconstruction module.

6.1 Cloud Database

User accounts are created and maintained under the cloud database. The system separately manages the user data and meta data values. Data values are stored in encrypted format. Query processing is performed under the cloud database environment.

6.2 Key Management

The key management process is used to maintain the key values. Multiple Crypto System (MCS) is used for the key management process. Separate key values are used for user data and meta data. Cloud database issues the key value to secure the query submission process.

6.3 Data Upload Process

Client application is used for the data upload process. Database tables are created and maintained in the data upload process. Meta data and records are encrypted before the upload process. Encrypted table data values are stored in the user storage area under the cloud database.

6.4 Privilege Management

User access permissions are assigned in the privilege management process. Privileges are assigned for the tables. Insert, delete, update and select permissions are used in the data sharing process. Re-encryption process is applied in the data sharing process.

6.5 Query Processing

Data download operations are carried out using query submission process. Encrypted query values are submitted to the cloud database. Cloud database prepare the query response and transfer to the client. Response data values are decrypted using the client secret key.

6.6 Database Reconstruction

Database tables are modified in the database reconstruction process. Column addition and deletion operations are supported in the reconstruction process. Concurrent user access is allowed in the database. Table data values are reassigned in the database manipulation process.

7 CONCLUSION

Cloud database services are integrated with data confidentiality and concurrent access models. Secure database as a service (Secure DBaaS) Framework is used to manage data access in encrypted cloud databases. The Secure DBaaS scheme is enhanced with data integrity features. Concurrent database structure modification and query security tasks are improved with security methods. The system eliminates the intermediate proxies in database management process. Database structure modification mechanism is adopted for multi user environment. The system improves the availability and scalability features. The response time in query processing is reduced by the system.

REFERENCES

[1] M.Armbrust et al.,"A View of Cloud Computing," Comm. of the ACM, pp. 50-58, 2010.

[2] W.Jansen and T. Grance, "Guidelines on Security and Privacy in Public Cloud Computing," Technical Report Special Publication 800-144, NIST, 2011.

[3] A.J. Feldman and E.W. Felten, "SPORC: Group Collaboration Using Untrusted Cloud Resources," Proc. Ninth USENIX Conf. Operating Systems Design and Implementation, Oct. 2010.

[4] V. Ganapathy and R. Motwani, "Distributing Data for Secure Database Services," Proc. Fourth ACM Int'l Workshop Privacy and Anonymity in the Information Soc., Mar. 2011.

[5] P. Mahajan and M. Walfish, "Depot: Cloud Storage with Minimal Trust," ACM Trans. Computer Systems, vol. 29, no. 4, article 12, 2011.

[6] "Oracle Advanced Security," Oracle Corpo ration,http://www.oracle.com/technetwork/data base /options/advanced-security, Apr. 2013.

[7] M. Hadavi, S. Cimato and Z. Ganjei, "AS5: A Secure Searchable Secret Sharing Scheme for Privacy Preserving Database Outsourcing," Proc. Fifth Int'l Workshop Autonomous and Spontaneous Security, Sept. 2013.

[8] R.A. Popa and H. Balakrishnan, "CryptDB: Protecting Confidentiality with Encrypted Query Processing," Proc. 23rd ACM Symp. Operating Systems Principles, Oct. 2011.

[9] Ferretti and Marchetti, "Supporting Security and Consistency for Cloud Database," Proc. Fourth Int'l Symp. Cyberspace Safety and Security, Dec. 2012.

[10] "Transaction Processing Performance Council," TPC-C, http://www.tpc.org, Apr. 2013.

Ms. S. Tharani received the B.Tech(IT) degree from Bannari Amman Intitute of Technology, Sathyamangalam, India in 2007-2010 and worked as lecturer in Shree Venkateshwara Hi-Tech Polytechnic College, Erode, India in 2011-2013pursuing ME(CSE) degree in Shree Venkateshwara Hi-Tech Engineering College, Erode, India in 2013-2015. Her research interests include Databases, network Security and datamining. She published 1 National Conferences, 1 Workshops.

Dr.T.Senthil Prakash received the Ph.D. degree from the PRIST University, Thanjavur, India in 2013 and M.E(CSE) degree from Vinayaka Mission's University, Salem , India in 2007 and M.Phil.,MCA.,B.Sc(CS) degrees from Bharathiyar University, Coimbatore India, in 2000,2003 and 2006 respectively, all in Computer Science and Engineering. He is a Member in ISTE New Delhi, India, IAENG, Hong Kong..IACSIT, Singapore SDIWC, USA. He has the experience in Teaching of 10+Years and in Industry 2 Years. Now He is currently working as a Professor and Head of the Department of Computer Science and Engineering in Shree Venkateshwara Hi-Tech Engineering College, Gobi, Tamil Nadu, and India. His research interests include Data Mining, Data Bases, Artificial Intelligence, Software Engineering etc.,He has published several papers in 17 International Journals, 43 International and National Conferences.

Mr.S.Prakadeswaran received the Bachelor of Engineering in Anna University,Chennai,Tamilnadu in 2008.He received the Master of Engineering in Anna University, Chennai, Tamilnadu in 2013. He is a Assistant professor in Shree Venkateshwara Hi Tech Engineering College, Gobichettipalayam, Tamilnadu. His research interest includes Wireless Networks and Pervasive computing