IPHV7I20021X
International Journal Of Advanced Research and Innovation -Vol.7, Issue .II
ISSN Online: 2319 – 9253
Print: 2319 – 9245

ACCESS CONTROL IN SOCIAL NETWORKS WITH ATTRIBUTE BASED ENCRYPTION TECHNIQUES

[#1] N.Yamini Gupta - M.Tech Pursuing,
[#2]M. Vasumathi Devi - Assistant Professor,
Department of Computer Science and Engineering,
Vignan's Nirula Institute of Technology and science for Women, Palakaluru, Guntur, AP.

Abstract: -A previously defined association between the attributes of a user and the cipher text associated. These attributes are exploited to determine the secret key of the user. A user with multiple attributes, the length of the key depends on the number of attributes. The existing methods that use reasonably computable decryption policy and procedures. The size of cipher text will vary linearly with the number of attributes. The cipher text remains constant in length, irrespective of the number of attributes. These attributes in a policy must be a subset of attributes in a secret key. When we use the access policy the attributes are used to generate the public key in order to encrypt the information and use the secret key decrypt the same information.

Keywords:- *Broadcast ,Access Control, Key Management, Data Security.*

I.INTRODUCTION

Cloud computing is drastically used in Information technology due to its parallel distributing services and low maintenance cost. Many organizations use Internet Service Providers as cloud which stores the sensitive data. All the employees of various organizations use cloud services to share, access, delete and upload the data securely. The Cloud Service Providers (CSP) such as Amazon, are providing the powerful datacenters to deliver various services to the cloud users. By storing all local data in the cloud, the employees can enjoy high-quality services in significant investments. The services offered by cloud providers are data storage. The business organizations allow the various department staff in same organization to store and share files in the cloud. So, the cloud service providers must provide the confidentiality while releasing the local data storage and maintenance. To provide the security to the cloud data, the best solution is encrypting the data files and then uploading it into the cloud. But designing the defensive distributing information scheme for groups in the cloud is a challenging issue.

Access policy is a mechanism used to provide the security facilities to the data. In traditional approach, the data is encrypted with the user's public key and then the file is uploaded into the cloud. Data outsourcing is actually relinquish user's crucial manage over the destiny of their information. The same as a outcome, the appropriateness of the information in the cloud is being put at possibility due to the next reason. Firstly, though the framework in the clouds is much greater and consistent than individual compute procedure, they are immobile face the wide range of both external and internal pressure for information reliability. Example of outages and protection breach of notable cloud service come out from time up to date. Next, these do survive various motivation for cloud service provider to perform falsely toward the cloud user as regards their outsource information kind. For example, Cloud Service Providers (CSP) might retrieve storage space for economic reason by throwing away information that have not been used or infrequently access, or constant hides information lost incident to sustain repute. In brief though outsourced information to the cloud is inexpensively gorgeous for long term large scale storage, it doesn't instantly propose any assurance on data reliability and accessibility. That complexity, if not accurately address, may obstruct the success of the cloud. Since users no longer acquire the storage of that information, conventional cryptographic primitive for the reason of information security defense can't be adopted straight. During downloads, all the information for its reliability and authentication is not a useful solution suitable to the expensiveness in Input-Output and communication Further, it is continually inadequate to determine the information correctness only while accessing the information, as it does not gives user accurateness maintain for that unaccessed information and may be too overdue to recovering the data lost or damages. Assume the huge volume of the outsource information and the users confine source capacity, the responsibilities of audit the information correctness in a cloud location should be horrible and more costly for the cloud user. As well, in the clouds of usage of cloud storage space can be decrease as possible, for this a user doesn't require to do many tasks in using the data. Specially, user might not go through the difficulty in verifying the data consistency. Additionally more number of users accesses the similar cloud storage space, articulate in activities settings. For easy managing that is attractive, this cloud only entertains authentication demand from a preferred party.

II.RELATED WORK

There are two ways of ABE depending on which secret keys or cipher texts that access policies are associated. In Key Policy Attribute-Based Encryption (KP-ABE) system , the cipher texts are labeled by the dynamic sender with the set of descriptive attributes, while the users private key is issued by the TAA(Trusted Attribute Authority) captures a policy, finally these type of cipher texts can decrypt the key. The KP-ABE scheme was suitable for organizations with rules to read particular documents. Typical applications of KP-ABE include secure forensic analysis and target broadcast [5]. For example, in a secure forensic analysis system, audit log entries could be annotated with attributes such as the name of the user, the date and time of the user action, and the type of data modified or accessed by the user action. While a forensic analyst charged with some investigation would be issued a private key that associated with a particular access constitution. The private key would only open audit log records whose attributes satisfied the access policy associated with the private key. The first KP-ABE creation was provided by Goyal et al. [5], which was very expressive in that it allowed the access policies to be expressed by any monotonic formula over encrypted data. The system was proved selectively secure under the Bilinear Diffie-Hellman assumption. Later, Ostrovsky et al. [6] proposed a KP-ABE scheme where private keys can represent any access formula over attributes, including non-monotone ones, by integrating revocation schemes into the Goyal et al. KP-ABE scheme. Bilinear Diffie-Hellman accomplishes this secure exchange by creating a “shared secret” (sometimes called a “Key Encryption Key” or KEK) between two devices. The shared secret then encrypts the symmetric key for secure transmittal. The symmetric key is sometimes called a Traffic Encryption Key (TEK) or Data Encryption Key (DEK). Therefore, the KEK provides for secure delivery of the TEK, while the TEK provides for secure delivery of the data itself.

EXISTING SYSTEM:

In the existing management system, multiple users may use different encrypting techniques according to their own ways using different cryptographic keys. Each user has keys from every owner. The owner maintains the user keys and details, since patients are always online. An alternative of this one is to employ the central authority but in this process we trust each and every owner on the central authority. The keys required to encrypt the data and decrypt the data under certain circumstances. The third party may access those keys include the business, who may want to access the employees private information.

PROPOSED SYSTEM:

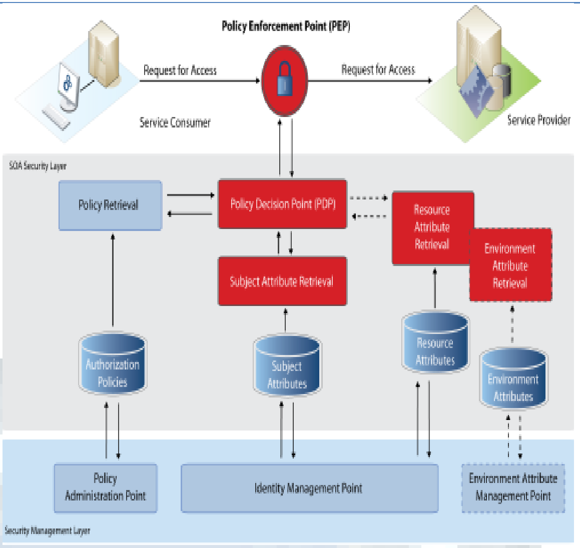
In proposed system, we are using Key Policy Attribute-Based Encryption (KP-ABE),cipher policy attribute based encryption and optimized security efficiency. The best example of this optimized method is health insurance portability and accountability acts as distribute file system in open network. An approach to entrance control in content sharing services is to authorize users to implement access controls on their information directly, rather than during a central administrator. However, this requires flexible and scalable cryptographic key management to support composite entire control policies. A local access control solution is to allocate one key for every user element, distribute the appropriate keys to users who have the subsequent attributes, and encrypt the media with the attribute keys repeatedly.

Algorithm

Encryption:
This algorithm takes a message *M*, the public key *PK*, and a set of attributes I as input. It outputs the cipher text *E* with the following format:
***E* = (*I*, ˜ *E*, {*Ei*}*i*)** where ˜*E = MY, Ei = Ti.*
Secret key generation:
This algorithm takes as input an access tree *T*, the master key *MK*, and the public key *PK*. It outputs a user secret key *SK* as follows.
***SK* = {*ski*}**
Decryption:
This algorithm takes as input the cipher text ***E*** encrypted under the attribute set **U**, the user’s secret key ***SK*** for access tree ***T***, and the public key ***PK***. Finally it output the message ***M*** if and only if **satisfies T**.

III.IMPLEMENTATION

Key to any successful ABAC implementation will be the Attribute Services (AS). AS makes attribute collection, dissemination, and security possible through the use of Policy Decision Points (PDPs) and Policy Enforcement Points (PEPs). For example, when a user tries to access a resource, the PEP will defer to the PDP, whose job is to decide whether or not to authorize the user based on the description of the user's attributes. Policies stored on the PEP and PDP systems provide the rule sets around which decisions are made.


Policy Enforcement Point (PEP)
Request for Access
Request for Access
Service Consumer
Service Provider
SOA Security Layer
Policy Retrieval
Policy Decision Point (PDP)
Resource Attribute Retrieval
Environment Attribute Retrieval
Subject Attribute Retrieval
Authorization Policies
Subject Attributes
Resource Attributes
Environment Attributes
Policy Administration Point
Identity Management Point
Environment Attribute Management Point
Security Management Layer

However, variations exist in attribute data due to a lack of process, training, typographical errors, etc. (i.e., officer rank may be encoded as "LtCol"or "Lieutenant Colonel"). This problem grows exponentially with the number of attributes and interconnected agencies, departments and systems. Additionally, making attribute data available externally can pose a significant security risk.

Disk encryption is a technology which protects information by converting it into unreadable code that cannot be deciphered easily by unauthorized people. Disk encryption uses disk encryption software or hardware to encrypt every bit of data that goes on a disk or disk volume. Disk encryption prevents unauthorized access to data storage. The term "full disk encryption" (or whole disk encryption) is often used to signify that everything on a disk is encrypted, including the programs that can encrypt bootable operating system partitions. But they must still leave the master boot record (MBR), and thus part of the disk, unencrypted. However there are hardware-based full disk encryption systems that can truly encrypt the entire boot disk, including the MBR.

Our work focuses on an important class of widely used applications that includes e-mail, personal financial management, social networks, and business tools such as word processors and spreadsheets. The following criteria define this class of applications:

- ➢ Provide services to a large number of distinct end users, as opposed to bulk data processing or workflow management for a single entity;
- ➢ Use a data model consisting mostly of sharable units, where all data objects have access control lists (ACLs) with one or more users; and
- ➢ Developers could run the applications on a separate computing platform that encompasses the physical infrastructure, job scheduling, user authentication, and the base software environment, rather than implementing the platform themselves.

1).Key Policy Attribute-Based Encryption (KP-ABE):

KP-ABE is a public key cryptography primitive for one-to-many communications. In KP-ABE, data are associated with attributes for each of which a public key component is defined. User secret key is defined to reflect the access structure so that the user is able to decrypt a cipher text if and only if the data attributes satisfy his access structure. A KP-ABE scheme is composed of four algorithms which can be defined as follows:

- Setup Attributes
- Encryption
- Secret key generation
- Decryption

Setup Attributes:

This algorithm is used to set attributes for users. From these attributes public key and master key for each user can be determined. The attributes, public key and master key are denoted as

Attributes- $U = \{1, 2. . . N\}$
Public key- $PK = (Y, T1, T2, . . . , TN)$
Master key- $MK = (y, t1, t2, . . . , tN)$

Encryption:

This algorithm takes a message M, the public key PK, and a set of attributes I as input. It outputs the cipher text E with the following format:

$$\boldsymbol{E = (I, \tilde{E}, \{Ei\}i)}$$

where $\tilde{E} = MY$, $Ei = Ti$.

Secret key generation:

This algorithm takes as input an access tree *T*, the master key *MK*, and the public key *PK*. It outputs a user secret key *SK* as follows.

SK = {ski}

Decryption:

This algorithm takes as input the cipher text ***E*** encrypted under the attribute set **U**, the user's secret key ***SK*** for access tree ***T***, and the public key ***PK***.

Finally it output the message ***M*** if and only if **U satisfies T**.

2) Proxy Re-Encryption (PRE):

Proxy Re-Encryption (PRE) is a cryptographic primitive in which a semi-trusted proxy is able to convert a cipher text encrypted under Alice's public key into another cipher text that can be opened by Bob's private key without seeing the underlying plaintext. A PRE scheme allows the proxy, given the proxy re-encryption key

rka↔b,

to translate cipher texts under public key ***pk1*** into cipher texts under public key ***pk2*** and vise versa.

3) Lazy re-encryption:

The lazy re-encryption technique and allow Cloud Servers to aggregate computation tasks of multiple operations. The operations such as

- Update secret keys
- Update user attributes.

IV. CP-ABE BASED SECURED CLOUD STORAGE ARCHITECTURE

In this section, first, we give a formal definition of our proposed scheme, and later we give the security model in which our scheme is proven to be secure.

4.1. System Description

Fig. 1 shows the architecture of the data sharing system, which consists of the following system entities:

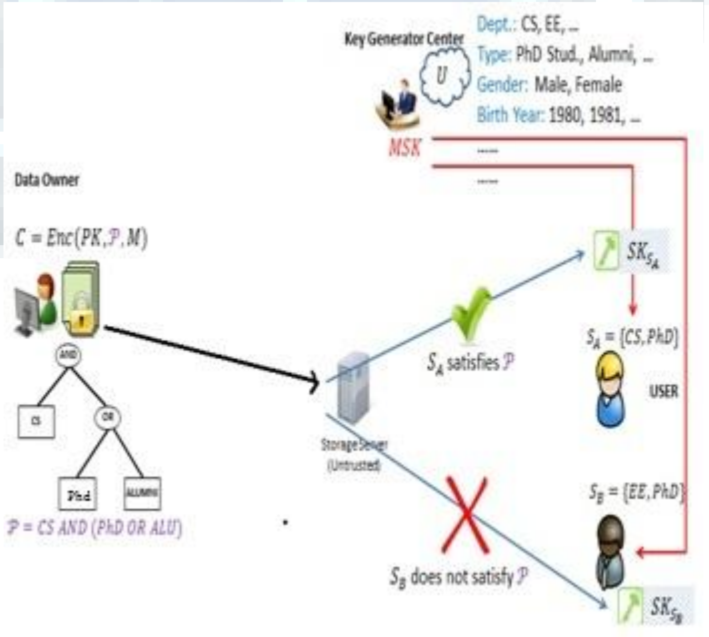


Figure 1. CP-ABE based Storage Architecture

Key generation center: It is a key authority that generates public and secret parameters for CP-ABE. It is in charge of issuing, revoking, and updating attribute keys for users. It grants differential access rights to individual users based on their attributes. It is assumed to be honest but curious. That is, it will honestly execute the assigned tasks in the system; however, it would like to learn information of encrypted contents as much as possible. Thus, it should be prevented from accessing the plaintext of the encrypted data even if it is honest.

Data storing center: It is an entity that provides a data sharing service. It is in charge of controlling the accesses from outside users to the storing data and providing corresponding contents services. The data storing center is another key authority that generates personalized user key with the KGC, and issues and revokes attribute group keys to valid users per each attribute, which are used to enforce a fine-grained user access control. Similar to the previous schemes [2],[3],[4], we assume the data storing center is also semi-trusted (that is, honest-but-curious) like the KGC.

Data owner: It is a client who owns data, and wishes to upload it into the external data storing center for ease of sharing or for cost saving. A data owner is responsible for defining (attribute- based) access policy, and enforcing it on its own data by encrypting the data under the policy before distributing it.

User: It is an entity who wants to access the data. If a user possesses a set of attributes satisfying the access policy of the encrypted data, and is not revoked in any of the valid attribute groups, then he will be able to decrypt the cipher text and obtain the data.

4.2. Cipher text-Policy Attribute-Based Encryption and Access Format

In our construction private keys will be identified with a set of descriptive attributes. A party that wishes to encrypt a message will specify through an access tree structure a policy that private keys must satisfy in order to decrypt. Each interior node of the tree is a threshold gate and the leaves are associated with attributes. A user will be able to decrypt a cipher text with a given key if and only if there is an assignment of attributes from the private key to nodes of the tree such that the tree is satisfied. We use the same notation as[5] to describe the access trees ,even though in our case the attribute s are used to identify the keys(as opposed to the data).specified in the private key, while the ciphertexts are simply labeled with a set of descriptive.

Access tree T structure : Let T be a tree representing an access structure. Each non-leaf node of the tree represents a threshold gate, described by its children and a threshold value. If num x is the number of children of a node x and k, x is its threshold value, then 0 <kx = numx. When kx = 1, the threshold gate is an OR gate and when kx = numx, it is an AND gate. Each leaf node x of the tree is described by an attribute and a threshold value kx = 1.

To facilitate working with the access trees, we define a few functions. We denote the parent of the node x in the tree by parent(x). The function att(x) is defined only if x is a leaf node and denotes the attribute associated with the leaf node x in the tree. The access tree T also defines an ordering between the children of every node, that is, the children of a node are numbered from 1 to num. The function index(x) returns such a number associated with the node x. Where the index values are uniquely assigned to nodes in the access structure for a given key in an arbitrary manner.

Satisfying an access tree: Let T be an access tree with root r. Denote by Tx the sub tree of T rooted at the node x. Hence T is the same as Tr. If a set of attributes Y satisfies the access tree Tx, we denote it as Tx(Y) = 1. We compute Tx(Y) recursively as follows. If x is a non-leaf node, evaluate Tx (Y) for all children x of node x. Tx(Y) returns 1 if and only if at least kx children return 1. If x is a leaf node, then Tx(Y) returns 1 if and only if att(x) Є Y.

Encrypt(PK,M,T) : The encryption algorithm encrypts a message M under the tree access structure T . The algorithm first chooses a polynomial qx for each node x (including the leaves) in the tree T . These polynomials are chosen in the following way in a top-down manner, starting from the root node R. For each node x in the tree, set the degree dx of the polynomial qx to be one less than the threshold value kx of that node, that is,dx = kx -1. Starting with the root node R the algorithm chooses a random s ? p and sets qR(0) = s. Then, it chooses Z dR other points of the polynomial qR randomly to define it completely. For any other node x, it sets qx(0) =qparent(x)(index(x)) and chooses dx other points randomly to completely define qx. Let, Y be the set of leaf nodes in T . The ciphertext is then constructed by giving the tree access structure T and computing

$$\mathbf{CT = (T\ ,\ C = Me(g,g)^{\alpha s}\ ,\ \ C = h^{s}\ ,}$$

$$\forall \mathbf{y} \in \mathbf{Y : Cy = g^{qy}\ ,Cy\ = H(att(y))^{qy(0)}\ ).}$$

KeyGen(MK,S). The key generation algorithm will take as input a set of attributes S and output a key that identifies with that set The algorithm first chooses a random r ? ,Z p and then random r ? Z for each attribute j ? S. Then it j p computes the key

$$\mathbf{SK = (D = g^{\ (\alpha+r)/\beta},}$$
$$\forall \mathbf{j} \in \mathbf{S : Dj= g^{r}\cdot\ H(j)^{rj},\ Dj= g^{\ rj}).}$$

Delegate(SK,S). The delegation algorithm takes in a secret key SK, which is for a set S of attributes, and another set S^1 such that S^1 Є S. The secret key is of the form SK = (D, ∀j ∈ S : Dj,D^1j). Then algorithm chooses random the r˜and r ˜k ∀ k Є S`. Then it creates a new secret key as

$$\mathbf{SK^{\sim}=(D^{\sim}= Df^{r}, \forall k\in S^{\sim}: Dk= Dkg^{r}H(k)^{rk}, D^{\sim}k= D^{-}kg^{rk})}$$

The resulting secret key SK is a secret key for the set S. Since the algorithm re-randomizes the key, delegated key is equivalent to one received directly from the authority.

<u>Cipher-Text Policy Attribute-Based Encryption</u> (CP-ABE)

In this Module, every user's personal secret key is generated with a set of attributes while every cipher text is connected with an access policy. A user effectively decrypts a cipher text only if his/her set of attributes satisfies the access policy specified in the cipher text. We briefly describe the CP-ABE.

We will extend this CP-ABE scheme to MCP-ABE scheme and use the latter in our access control scheme.

- *<u>AB-Setup</u>*

  It is an initialization algorithm run by an Attribute Authority (AA). It takes as input a security and outputs a Public Key (PK) and a Master Key(MK).

- *<u>AB-Keygen</u>*

  It is run by AA to issue a personal secret key to a user. It takes as input MK and the set of attributes A of the user, and outputs the private secret key SK connected with specifically, for each user.

- *<u>AB-Encrypt</u>*

  Data owner to encrypt a message according to an access tree.

- *<u>AB-Decrypt</u>*

  Data consumer in control of a set of attributes A and the secret key SK in order to decrypt the cipher-text CT with an access policy.

IV.CONCLUSION

In the proposed work, it is mandatory to provide the security to data in personal/private involves in online nature. Some of the data centers are utilizes the high-end protection systems to provide the security to the data. we know that the data centers are nothing but a cloud adding protection to the single cloud it leads to all services provided by client means thousands of services are benefited and terabytes of client data are protected.

REFERENCES

[1] [H. Abu-Libdeh, L. Princehouse and H. Weatherspoon, "RACS: a case for cloud storage diversity", SoCC'10:Proc. 1st ACM symposium on Cloud computing, 2010, pp. 229-240.
[2] P. Ammann and S. Jajodia, "Distributed Timestamp Generation in Planar Lattice Networks," ACM Trans. Computer Systems, vol. 11, pp. 205-225, Aug. 1993.
[3] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted
[4] Stores," Proc. ACM Conf. Computer and Comm. Security, pp. 598-609, 2007.
[5] E. Barka and A. Lakas, "Integrating Usage Control with SIP-Based Communications," J. Computer Systems, Networks, and Comm.,vol. 2008, pp. 1-8, 2008.
[6] D. Boneh and M.K. Franklin, "Identity-Based Encryption from the Weil Pairing," Proc. Int'l Cryptology Conf. Advances in Cryptology, pp. 213-229, 2001.
[7] R. Bose and J. Frew, "Lineage Retrieval for Scientific Data Processing: A Survey," ACM Computing Surveys, vol. 37, pp. 1-28, Mar. 2005.
[8] P. Buneman, A. Chapman, and J. Cheney, "Provenance Management in Curated Databases," Proc. ACM SIGMOD Int'l Conf.
[9] Management of Data (SIGMOD '06), pp. 539-550, 2006.
[10] B. Chun and A.C. Bavier, "Decentralized Trust Management and Accountability in Federated Systems," Proc. Ann. Hawaii Int'l Conf. System Sciences (HICSS), 2004.

[11] C. Dwork, “The Differential Privacy Frontier Extended Abstract,” *Proc. 6th Theory of Cryptography Conf.* (TCC 09),LNCS 5444, Springer, 2009, pp. 496-502.
[12] C. Gentry, “Fully Homomorphic Encryption Using Ideal Lattices,” *Proc. 41st Ann. ACM Symp. Theory Computing* (STOC 09), ACM, 2009, pp. 169-178.
[13] E. Naone, “The Slow-Motion Internet,” *Technology Rev.*, Mar./Apr. 2011; www.technologyreview.com/files/54902/
[14] GoogleSpeed_charts.pdf.Greenberg, “IBM’s Blindfolded Calculator,” *Forbes*,13 July 2009; www.forbes.com/forbes/2009/0713/breakthroughs-privacy-super-secret-encryption.html.
[15] Cloud Data Protection for the Masses , Dawn Song, Elaine Shi, and Ian Fischer, *University of California, Berkeley* Umesh Shankar, *Google*, 2012 IEEE Published by the IEEE Computer Society JANUARY 2012
[16] Abraham, G. Chockler, I. Keidar and D. Malkhi, "Byzantine disk paxos: optimal resilience with Byzantine shared memory", Distributed Computing, 18(5), 2006, pp. 387-408.

AUTHORS PROFILE:

[1] **N.Yamini Gupta** received the B.Tech degree in Information Technology from Vignan’s Nirula Institute of Technology and science for Women, Palakaluru, Guntur D’t,, AP., in 2012 from JNTUK.She is currently a masters student in the department of Computer Science and Engineering at Vignan’s Nirula Institute of Technology and science for Women, Palakaluru, Guntur D’t, AP.

[2]. **M. Vasumathi Devi** possessed her B.Tech(C.S.E) in 2003 from JNTU and M.Tech(C.S) in 2010 from JNTUK. She worked as an Assitant professor in SSN engineering college at ongle, VIGNAN vadlamudi at Guntur, and now she is working as Asst. Professor in Department of Computer Science & Engineering at vignan’s nirula institute of technology and science for women, Palakaluru, Guntur D’t , AP. She has nine years of teaching experience . She has guided many projects in the area of Computer Networks, Data Mining, and Image processing for CSE & IT Departments. She attended for a workshop on Unified Modeling Language conducted in the year 2005 at Kottam Tulasi Reddy engineering college at Mahabubnagar. She had attend many National Conferences. She is persuing her Ph.D. Research work in Wireless Sensor Networks.