ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

Constant Size Cipher text Attribute-Based Encryption with Lively Membership

Saranya P.P[1], Alphonsa Kuriakose[2]

M.Tech Student, Dept. of CSE, VJCET, M G University, Vazhakulam, Kerala, India[1]

Assistant Professor, Dept. of CSE, VJCET, M G University, Vazhakulam, Kerala, India[2]

ABSTRACT: Attribute-based encryption provides privacy protection for the users by a set of attributes. An encryptor can select a set of attributes to encrypt the secret message. Those users whose attribute values match with the conditions on selected attribute values can only decrypt the message. There are various issues involved in attribute-based encryption. The key factor in attribute-based encryption is the attributes itself. All attribute values need to be up to date and correct. But this is a difficult task because the attribute values associated with the users may change anytime. Also users may leave from the group or new users may join the system. To keep the attribute values correct, a lively membership mechanism is introduced. Also one of the drawbacks of attribute-based encryption is the linearly increasing cipher text size. To avoid this, constant cipher text scheme is introduced which will make the cipher text size constant with any given number of attributes. The performance evaluation shows that the scheme provides an efficient ABE with lively membership and constant size cipher text.

KEYWORDS: Attribute-Based Encryption; Cipher text; Cipher text-Policy Attribute-Based Encryption; Lively Membership; Constant Size Cipher text.

I. INTRODUCTION

Attribute-based encryption, from the name itself is an encryption technique based on a set of attributes. The attribute values owned by the users are used to specify the access policies to control access to the messages. The access policies in attribute-based encryption can be of two types: key-policy and cipher text-policy. Key-policy schemes specify the access policy in the user's private keys. But in cipher text-policy schemes, the access policy is included in the cipher text.

Attribute-based encryption can be used in different real life situations. Consider the case of a professional community. The community includes different employees with equal priorities. They want to share some files and messages. The employees can be from different offices, departments, locations, and of holding different positions. They don't know each other. In this community, if one user wants to share something, he can select a set of attributes so that only the other users with these attribute values can access the information. The scheme can be used to ensure fine grained access control by the set of attribute values owned by the users. Data confidentiality is provided by encrypting the message before sending it to the group. New users can join to the group anytime and after joining they can share information by specifying different access policies. Users can quit from the system anytime and after that they won't be able to receive any messages.

Attribute-based encryption can be classified into two types.

- Key-Policy ABE (KP-ABE).
- Cipher text-Policy ABE (CP-ABE).

Key-Policy ABE: The access policies are associated with user's private keys. Private key is generated based on the attribute values owned by the user. Thus in KP-ABE, the access policy is fixed and is independent of the cipher text.

Cipher text-Policy ABE: Access policies are associated with the cipher text. CP-ABE is more flexible compared to KP-ABE because in second one, access policy is specified when the cipher text is generated. This paper deals with the issues in CP-ABE.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

II. RELATED WORK

Attribute-based encryption is an encryption technology where the privacy of messages is protected by a set of attributes. CP-ABE was first introduced by Amit Sahai and Brent waters [3]. It has many applications. One of the uses of attribute-based encryption is to provide fine-grained data access control for wireless sensor networks [4]. Another use is to provide channel access control in pay-TV broadcasting system using attribute-set-based encryption [5]. CP-ABE scheme with single authority takes as input single authority key and plaintext. Multiple-authority keys CP-ABE was introduced in [6]. One of the problems with attribute-based encryption, the key escrow problem in which private keys associated with all users are known by the key generation centre was addressed by [7].

In attribute-based encryption, ownership of decryption could be possessed by multiple users. In order to solve this, Traceable CP-ABE (T-CP-ABE) system was developed in [8]. One of the disadvantages of existing ABE schemes is that decryption includes expensive pairing procedures. Outsourced decryption that eliminates the decryption overhead was proposed in [9]. In order to reduce trust on central authority, decentralized CP-ABE was introduced in [10]. CP-ABE schemes have long decryption keys. As a solution, a new CP-ABE scheme with fixed size decryption keys was proposed in [11]. In order to reduce the complexity of key issuing and decryption, secure outsourced ABE system was proposed which supports both outsourced key-issuing and decryption [12]. For providing the correctness of user's attributes, a dynamic membership was introduced which handles user's attribute values by several algorithms [1]. The existing CP-ABE schemes results in very large cipher text size. A constant CP-ABE scheme was introduced in [2] that significantly reduce the cipher text to fixed size with any number of attributes.

III. SYSTEM OVERVIEW

The advanced encryption technique known as attribute-based encryption (ABE) provides privacy protection for the users by the attributes. In this type of encryption, the encrypted secret data can only be decrypted by the users who meet the specified constraints. Out of the two available ABE schemes, CP-ABE is more flexible. So this paper deals with issues in CP-ABE scheme. Various improvements in cipher text-policy attribute-based encryption such as multiple authority keys, secure key-issuing protocol, traceability, outsourced decryption, constant-size keys and checkability are provided by [6], [7], [8], [9], [11], [12].

The base of attribute-based encryption is the attributes itself. Most important factor in attribute-based encryption is the user's attribute values. So to keep attribute-based encryption efficient and privacy preserving, prime importance is given for maintaining the correctness of user's attribute values as explained in [1]. Taking this into consideration, this paper proposes a lively membership mechanism which will keep the user's attribute values correct by using several algorithms. It provides the provision for the user to update the attributes and can leave from the system and also the facility to re-join anytime. The users who are currently inactive in the system cannot be able to read any messages even though his attributes match with the conditions in the access policy. CP-ABE scheme incurs a bulky cipher text. As a solution, linearly increasing cipher text size is reduced to a fixed size by constant size cipher text method as explained in [2]. The scheme is shown in Fig. 1.

Overall this paper deals with the following:

- Lively Membership: Attributes can be maintained independently; users can join and leave anytime.
- Constant Size Cipher text: Size of the generated cipher text remains fixed disregarding the number of attributes.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

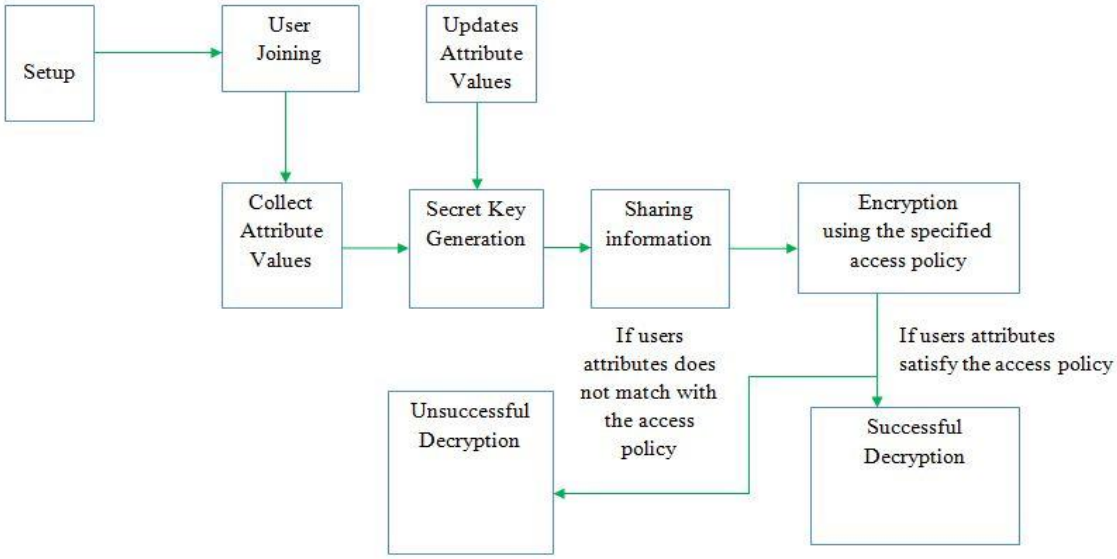

Fig. 1. ABE Scheme with Constant Size Cipher text and Lively Membership

The different modules in this scheme are:

- Setup
- User Joining
- User Leaving
- Updating Attribute Values
- Encryption
- Decryption

A. Setup:

This algorithm involves setting a background for attribute-based encryption. This will take as input the total number of attributes in the system. It returns the public key and master key for the system. If there are k attributes in the system, each attribute can take three values: present, absent and not considered. So the total number of attribute values is 3 times the total attributes since each attribute can take 3 values. Attribute-based encryption is a type of pairing based cryptography. So the first step is to generate two bilinear groups with prime order p. Order indicates the total number of elements in the group. Then, select some random generator from group. In third step, public key is generated using these values. Then select a random number for master key and generate the master key. Finally the algorithm will output the private key and master key for the system.

Algorithm Setup

Input: Number of attributes in the system, k.
Output: Public key and master key for the system.

Step 1: Select two bilinear groups of prime order p.
Step 2: Select a random generator g from bilinear group.
Step 3: Calculate public key (PK) for the system using random generator.
Step 4: Take a random element α from the set of integers Z_p.
Step 5: Select a random element $\gamma \in Z_p$ and set $m = g^{\gamma}$

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

Step 6: The master key is $MK = \{\gamma, \alpha\}$.

B. User Joining:

This algorithm is executed each time a new user joins the system. Each user is associated with a collection of attributes. When a user joins the system, particular attribute values for the user are collected. Based on the attribute values, the private key of the user is generated. First select k random numbers and calculate its sum. Compute value D using the given equation. Then calculate D_i for each user by taking the attribute list associated with the user. Compute value $D_i^{'}$ for each user. Then the newly enrolled user is added to the existing set of users. Finally the private key of the user is computed and output to the user.

Algorithm User Joining

Input: User s with attribute list L_s .

Output: The private key of user.

Step 1: Choose k random numbers $\{r_i\}_{i\in[1,k]}$ from the set Z_p .

Step 2: Compute the sum of k random numbers, r .

Step 3: Calculate a value $D = m^r$.

Step 4: Calculate D_i for each user by taking the attribute list of user, random generator, master key and r_i .

Step 5: Calculate $D_i{}'$ using random generator, master key and random number r_i .

Step 6: Add user s to the existing set of users $U = U \cup \{s\}$

Step 7: The private key for user is computed as: $SK = \left(D, \{D_i\}_{\forall i\in[1,k]}, \{D_i^{'}\}_{\forall i\in[1,k]}\right)$

C. User Leaving:

User leaving algorithm is invoked when a user wants to leave from the system. The public parameter is generated in the setup phase using all the user's attribute values. When a user leaves, first step is to update the public parameter PK. In order to keep the public parameter up to date, the version number of the public parameter is updated. After these operations, the particular leaved user is removed from the set of registered users. If some registered user s is revoked by the system, the following algorithm is invoked.

Algorithm User Leaving

Input: The user id of user s.

Output: The updated public key.

Step 1: Update the public parameter PK.

Step 2: Set user status as inactive.

Step 3: Increase version number of public parameter.

Step 4: Set $U = U \setminus \{s\}$.

D. Updating Attribute Values

Some user's attribute values may be changed over time. So there is a need to update the user's attribute values in order to keep the system efficient and privacy-preserving. When updating algorithm is invoked the user's attribute value is changed to a new value as given by the user. The private key of the user is also updated accordingly. For generating the updated private key, first randomly choose k numbers. Then compute the private key using the updated attribute values. Also, when the attribute values are changed, the public key of the system is also changing. So increase the version number and update public parameter PK accordingly. Finally the algorithm will output the updated private

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

key and public key. When a registered user applies for updating her/his attribute to a new value, the following algorithm is invoked.

Algorithm Updating Attribute Values

Input: The attribute need to be updated and new value for the attribute.
Output: The updated private key and public key.

Step 1: Randomly pick k numbers $\{r_i^{'}\}_{\forall i \in [1,k]}$.

Step 2: Compute the sum $r^{'} = \sum r_i^{'}$.

Step 3: Calculate $D^{'} = m^{r^{'}}$.

Step 4: Calculate $D_i^{'}$ using the updated attribute list.

Step 5: Calculate $D_i^{''}$ using updated value.

Step 6: Compute private key by using the updated attribute values.
Step 7: Increase version number and update PK.

E. Encryption

In cipher text-policy ABE the access policy for the message is specified in the cipher text. Constant size cipher text-policy attribute-based encryption retains the cipher text size constant with any number of attributes. Every enrolled user in the system is associated with an attribute list L. This will specify for each user whether the particular attribute value is present or absent. Each attribute can take two values: either the user is having that attribute value, or he is not having that attribute value.

In order to generate the cipher text, first select a random element t. Then generate one-time symmetric encryption key. Particular message is encrypted using this symmetric encryption key. Then specify an access policy P. The access policy specifies which of the attribute values are needed for decrypting this cipher text. For generating the access policy, this will take every attribute value in the system and specify whether this attribute value is present, absent or not considered. This is specified for every attribute value in the system. So for a particular user, if he/she is having the attribute value in the access policy can only access the particular message. Then specify the message header. The next step is to make the access policy unidentified to keep it secret. The cipher text is a combination of encrypted message, message header and the unidentified access policy.

Algorithm Encryption

Input: Public key PK, Message M and Access policy P.
Output: Cipher text CT.

Step 1: Select a random element t from the set of integers Z_p.

Step 2: Calculate symmetric encryption key using first and k-th elements of public key.
Step 3: Encrypts the message using symmetric key.

Step 4: Specify the access policy P with k attributes, $P = \{P[i]\}_{i \in [1,k]}$, $P[i] \in \{V_i^+, V_i^-, V_i^*\}$

If i-th attribute value is denoted as V_i, then V_i^+ indicates the attribute value is present.

V_i^- and V_i^* indicates attribute value is absent and not considered respectively.

Step 5: Specify the message header, Header.

Step 6: Generate unidentified access policy, $\overline{P} = P \cap \{V_i^*\}_{i \in [1,k]}$

Step 7: The cipher text is, $CT = (\overline{P}, \{M\}_{Key}, Header)$

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

F. Decryption

The user can decrypt the cipher text only if the attribute values specified in the access policy matched with the attribute values of particular user. The first step is to replace the attributes in the access policy with the particular attribute values of the user. If the user's attribute values are correct, the procedure will produce the correct value. And the user can decrypt the cipher text successfully. Otherwise the decryption will fail and the user is not able to access the particular message. The user can only decrypt the cipher text if access policy P is subset of L.

Algorithm Decryption

Input: Public key PK, the private key of the user and the cipher text CT.
Output: If the user's attributes satisfy the access policy then the valid message, otherwise a random string.

Step 1: Construct an approximate access policy: Q
Step 1.1: Approximate access policy is constructed by replacing hidden attributes in the access policy with the attribute values of particular user.
Step 2: For $\forall i \in [1,k]$, user s calculates T_1 and T_2 as follows:
Step 2.1: Calculate T_1 using values in the message header.
Step 2.2: If access policy Q is subset of attribute set L_s, then
Step 2.2.1: s calculates: T_2 using value in the message header and element D_i in private key.
Step 2.3: Else if access policy Q is subset of don't care values, then
Step 2.3.1: s calculates: T_2 using value in the message header and $D_i^{'}$ in private key.
Step 3: Get key using T_1 and T_2.
Step 4: Decrypt the message using the key.
Step 5: If receiver satisfies the access policy,
Step 5.1: Decryption will succeed and user gets the valid message M.
Step 5.2: Otherwise decryption will fail and user gets a random string.

Most important in attribute-based encryption is to keep the user's attribute values correctly. The above six algorithms provide a lively membership mechanism which will maintain the user's attribute values properly. The encryption algorithm will always produce an efficient cipher text size which will not increase linearly with the count of attributes in the access policy.

IV. PERFORMANCE EVALUATION

In cipher text-policy attribute-based encryption, the access policy is specified in the cipher text and is decided when the cipher text is generated. The access policy is specified in different ways. In one method, the access policy is specified as a tree structure. The tree is having root and leaf nodes. The leaf nodes of the tree are associated with attributes and specified values for the attributes. Each node of the tree is associated with a polynomial function which is used for specifying the access policy. If the access policy is specified as a tree structure, size of the cipher text increases consecutively with the count of attributes in the access policy. So the cipher text becomes bulky and the scheme is inefficient. In order to avoid this, a constant size cipher text scheme is used. The cipher text size remains constant with any number of attributes. The access policy is not specified as a tree structure. Instead, it is specified as whichever attribute values are present, absent and not considered.

A. Encryption Using Access Tree Structure Method:

The following procedure is used to encrypt a message:

- Specify an access tree structure, leaf nodes of the tree correspond to the attributes and values.
- Access policy is specified by taking the leaf node values of the access tree structure.
- The secret message is encrypted using a symmetric key.
- Access policy for the message is specified using the above tree structure.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

B. Encryption Using Constant Size Cipher text Method:

For making the cipher text constant, the tree structure is removed and instead the access policy is specified using +, - and * to indicate which attribute values are present, absent and not considered. The encryption algorithm will take as input public key PK, message M and access policy P and outputs the Cipher text CT. Encryption method is as follows:

- Select a random element t from set of integers and calculate symmetric encryption key.
- Encrypts the message using symmetric key.
- Specify access policy P with k attributes and specify the message header, Header.
- Outputs the cipher text CT.

The above method generates a cipher text whose size remains constant with any given number of attributes.

The execution time of encryption for both methods is estimated in milliseconds for different access policies. The values are shown in Table I.

TABLE I
EXECUTION TIME OF ENCRYPTION IN MILLISECONDS FOR DIFFERENT ACCESS POLICIES

Access Policies	Execution Time of Encryption	
	Access Tree Structure Method	Constant Size Cipher text Method
Access Policy 1	1560	210
Access Policy 1	1562	234
Access Policy 1	1770	297
Access Policy 1	1785	299

The graph comparing the execution time of encryption in milliseconds for both methods is shown in Fig. 2. The execution time is estimated for different access policies starting from simple access policy to most complex access policies and compared. In all cases, the execution time of encryption is less for constant size cipher text method.

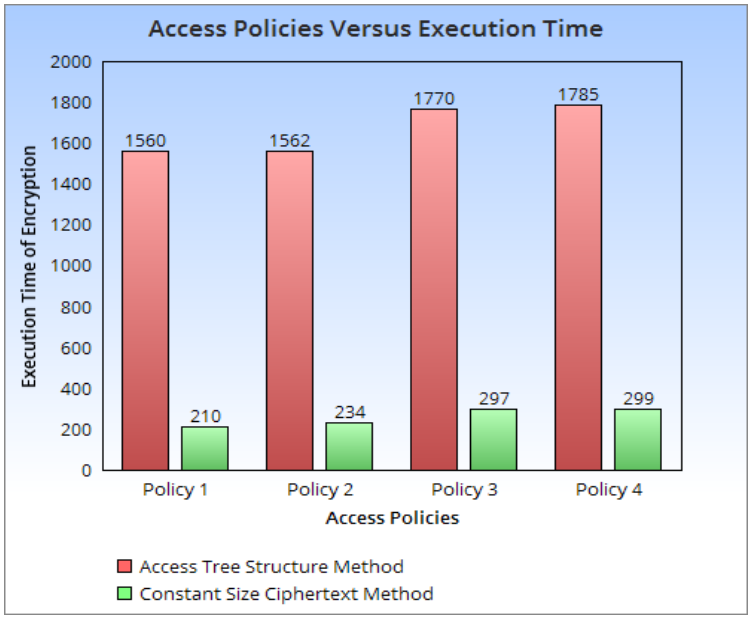

Fig. 2. Graph Comparing the Execution Time of Encryption

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

V. CONCLUSION AND FUTURE WORK

Attribute-based encryption is an encryption technology, where the privacy of the information is protected by a set of attributes. In cipher text-policy attribute-based encryption, cipher text involves some access policies and only the users with those attribute values in the access policy can decrypt the cipher text. The work considered various issues and their solutions involved in attribute-based encryption. Most of the schemes focused on solving issues like key escrow problem, expensive decryption operations, traceability, checkability etc. The management of attributes is important in attribute-based encryption. Lively membership management is able to handle the attributes properly. To make cipher text-policy attribute-based encryption more efficient, the progressively increasing cipher text size is reduced to fixed size by a constant size cipher text mechanism. In future, it is possible to extend the scheme to provide more security by a secure key issuing mechanism using multiple authorities. Another direction for future research is to make the scheme more efficient by shorter private keys.

REFERENCES

1. Chun-I Fan, Vincent Shi-Ming Huang, and He-Ming Ruan, "Arbitrary-State Attribute-Based Encryption with Dynamic Membership," IEEE Trans. Computers, vol. 63, no. 8, Aug. 2014.
2. Zhibin Zhou, Dijiang Huang, and Zhijie Wang, "Efficient Privacy-Preserving Cipher text-Policy Attribute-Based Encryption and Broadcast Encryption," IEEE Trans. Computers, vol. 64, no. 1, Jan. 2015.
3. J. Bethencourt, A. Sahai, and B. Waters, "Cipher text-Policy Attribute- Based Encryption," Proc. IEEE Symp. Secur. Privacy, pp. 321-334, 2007.
4. S. Yu, K. Ren, and W. Lou, "FDAC: Toward Fine-Grained Distributed Data Access Control in Wireless Sensor Networks," IEEE Trans. Parallel and Distributed Systems, vol. 22, no. 4, pp. 673-686, Apr. 2011.
5. ZhiguoWan, June Liu, Rui Zhang, and Robert H. Deng, "A Collusion-Resistant Conditional Access System for Flexible-Pay-Per-Channel Pay-TV Broadcasting," IEEE Trans. Multimedia, vol. 15, no. 6, Oct. 2013.
6. Xiaoyuan Yang, Weiyi Cai, and Ping Wei, "Multiple-Authority Keys CP-ABE," Proc. Third IEEE Intl Conf. Communication Software and Networks, 2011.
7. Junbeom Hur, "Improving Security and Efficiency in Attribute-Based Data Sharing," IEEE Trans. Knowledge and Data Engineering, vol. 25, no. 10, Oct. 2013.
8. Zhen Liu, Zhenfu Cao, and Duncan S. Wong, "White-Box Traceable Cipher text-Policy Attribute-Based Encryption Supporting Any Monotone Access Structures," IEEE Trans. Information Forensics and Security, vol. 8, no. 1, Jan. 2013.
9. Junzuo Lai, Robert H. Deng, Chaowen Guan, and Jian Weng, "Attribute-Based Encryption Verifiable Outsourced Decryption," IEEE Trans. Information Forensics and Security, vol. 8, no. 8, Aug. 2013.
10. Jinguang Han, Willy Susilo, Yi Mu, Jianying Zhou, and Man Ho Au, "Improving Privacy and Security in Decentralized Cipher text-Policy Attribute-Based Encryption," IEEE Trans. Information Forensics and Security, Aug. 2014.
11. Fuchun Guo, Yi Mu, Willy Susilo, Duncan S. Wong, and Vijay Varadharajan, "CP-ABE with Constant-Size Keys for Lightweight Devices," IEEE Trans. Information Forensics and Security, vol. 9, no. 5, May. 2014.
12. Jin Li, Xinyi Huang, Jingwei Li, Xiaofeng Chen, and Yang Xiang, "Securely Outsourcing Attribute-Based Encryption with Checkability," IEEE Trans. Parallel and Distributed Systems, vol. 25, no. 8, Aug. 2014.

BIOGRAPHY

Saranya P.P is an M.Tech student in the Computer Science and Engineering Department, Viswajyothi College of Engineering and Technology, Mahatma Gandhi University. She received Bachelor of Technology (B.Tech) degree in 2013 from Viswajyothi College of Engineering and Technology, Vazhakulam, Mahatma Gandhi University, India. Her research interests are Information Security, Data Mining etc.

Alphonsa Kuriakose is an Assistant Professor in the Computer Science and Engineering Department, Viswajyothi College of Engineering and Technology, Mahatma Gandhi University. She received Bachelor of Technology (B.Tech) degree in 2010 from Viswajyothi College of Engineering and Technology, Vazhakulam, Mahatma Gandhi University, India and her Master of Technology (M.Tech) degree in 2012 from Adi Shankara Institute of Engineering and Technology, Kalady, Mahatma Gandhi University, India. Her research interests are Computer Networks, Information Security, Digital Image Processing etc.