ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

A Verifiable Cloud Storage using Attribute Based Encryption and Outsourced Decryption with Recoverability

R.V.Agalya, K.Karthika Lekshmi
ME Computer and Communication (II year) Cape Institute of Technology Tirunelveli,
Assistant Professor- Dept.of IT Cape Institute of Technology Tirunelveli

Abstract*— Cloud shares infrastructure between several organizations and it is managed internally or by a third-party. The user stores the data in an encrypted format. ABE is an encryption scheme used by the user to store the data in the cloud. ABE is a public-key based one to many encryption techniques which allows users to encrypt and decrypt data based on user attributes. Access control of encrypted data stored in the cloud is, by using access polices and ascribed attributes associated with private keys and cipher texts. In existing ABE schemes decryption has expensive paring operations and the complexity of the access policy is proportional to the number of attributes. An ABE system with outsourced decryption eliminates the decryption overhead. Here user provides data to the cloud service provider, with a transformation key that allows the cloud to translate any ABE cipher text satisfied with the user's attributes or access policy into a simple cipher text. In this paper, an ABE encryption and outsourced decryption with verification and recovery technique is proposed. This technique effectively secures the data and also provides the correctness of the retrieved data along with the recovery mechanism for the transmitted data in case of malicious attack. The implementation of this scheme will show the correctness of the secure data storage and the recovery process.

Index Terms— **Malicious attack, Attribute based Encryption, Verifiability, Recoverability.**

I. INTRODUCTION

Internet technology is growing quickly, and people can process, store, or share with their data by using its ability. Recently, the cloud has emerged to provide various application services to satisfy user's requirement. Cloud computing provides the tools and technologies to build data/compute intensive parallel applications with much more affordable prices compared to traditional parallel computing techniques. It has three servicing models Infrastructure as a Service, Software-as-a-Service, and Platform as a Service. Saas type of cloud computing allow users to run existing online applications. It delivers a single application through the browser to thousands of customers using a multitenant architecture. PaaS allow users to run their own applications using supplier specific tools and languages. It comprises the environment for developing and provisioning cloud applications. The users of this layer are developers seeking to develop and run a cloud application for a particular platform. Iaas services on the infrastructure layer allow user to run any application they please on cloud hardware on their own choice. IT resources that are combined under the heading Infrastructure-as-a-Service (IaaS) include services linked to computing resources, data storage resources, and the communications channel. In the storage service application, the cloud can let the user, data owner to store his data, and share this data with other users via the cloud, because the cloud can provide the pay as you go environment where people just need to pay the money for the storage space they use. For protecting the confidentiality of the stored data, the data must be encrypted before uploading to the cloud. The encryption scheme used is attribute-based encryption. The ABE scheme used a user's identity as attributes, and a set of attributes were used to encrypt and decrypt data. One of the main efficiency drawbacks of the most existing ABE schemes is that decryption is expensive for resource-limited devices due to pairing operations, and the number of pairing operations required to decrypt a cipher text grows with the complexity of the access policy. The ABE scheme can result the problem that data owner needs to use every authorized user's public key to encrypt data. Key-policy attribute-based encryption (KP-ABE) scheme built the access policy into the user's private key and described the encrypted data with user's attributes. The KP-ABE scheme can achieve the grained access control and more exibility to control users than ABE scheme. But the disadvantage of KP-ABE is that the access policy is built into an user's private key, so data owner can't choose who can decrypt the data except choosing a set of attributes which can describe this data. And it is unsuitable in certain application because a data owner has to trust the key issuer. CP-ABE scheme built the access policy into the encrypted data; a set of attributes is in a user's key. The CP-ABE scheme addresses the problem of KP-ABE that data owner only trusts the key issuer [13]. To assess the performance of our ABE scheme with verifiable outsourced decryption, we implement the CP-ABE scheme with verifiable outsourced decryption and conduct experiments.

II. RELATED WORKS

As a lot of sensitive information is shared and hold on by third-party sites on the net, there'll be a desire to cipher information hold on at these sites. One disadvantage of encrypting information is that it will be by selection shared solely at a coarse-grained level (i.e., giving another party your personal key). Goyal et al [2] proposed a scheme for fine-grained sharing of encrypted information that it has the tendency to developed Key-Policy Attribute-Based coding. In that, attributes and personal keys are related to access structures that manage the cipher texts that the user is ready to rewrite. It didn't hide the set of attributes underneath that the information is encrypted. Cheung et al [1] proposed a ciphertext policy attribute-based coding (CP-ABE), every secret key is associated with a set of attributes, and each ciphertext is associated with access structure on attributes. Secret writing is enabled if and only if the user's attribute set satisfies the ciphertext access structure. This provides fine-grained access management on shared information in several sensible settings, likewise as secure databases and secure multicast. It gifts a variant with well smaller ciphertexts and faster encryption/decryption operations. The most arrange is to form a hierarchy of attributes, so fewer cluster components square measure required to represent all attributes within the system. This economical variant is proven to be controller secure. Herranz et al [5] proposed the first attribute-based encryption (ABE) schemes allowing for truly expressive access structures and with constant ciphertext size. First it results a ciphertext-policy attribute-based encryption (CP-ABE) scheme with O(1)-size ciphertexts for threshold access policies and where private keys remain as short as in previous systems. As a second result, a certain class of identity-based broadcast encryption schemes generically yields monotonic key-policy attribute-based encryption (KP-ABE) systems in the selective set model. Waters et al [4] proposed ciphertext-policy attribute based encryption. to kept encrypted data confidential even if the storage server is untrusted. It is secure against collusion attacks. Previous ABE systems used attributes to describe the encrypted data and built policies into user's keys and a party encrypting data determines a policy for who can decrypt. Thus, their methods are conceptually closer to traditional access control methods such as Role-Based Access Control (RBAC). Raykova et al [3] proposed the VC scheme that verifies any function in the class of functions covered by the permissible ABE policies (currently Boolean formulas). It is very efficient verification algorithm that depends only on the output size.[7] presented a multi-function VC scheme allows the verifiable evaluation of multiple functions on the same pre-processed input. Green et al [12] proposed Outsourced decryption; it has two keys called secret key and transformation key. Proxy will be able to transform any ABE ciphertext into a short ciphertext for the user. While the security definitions show that an attacker will not be able to learn an encrypted message, there is no guarantee on the transformation's correctness. Junzuo et al [20] proposed a scheme to verify the retrieved content. It just gives a result whether the content is original or modified. It does not specify where the data get modified. If it results the content is modified then the data cannot be used without knowing the where the modification is present.

III. PRELIMINARIES

A. Access Tree

Let T be a tree representing an access structure. Each non-leaf node is described by its children and a threshold value. Non-leaf nodes of the tree represented as threshold gate. If num_x.is the number of children of a node x and k_x is its threshold value, then $0 < k_x \leq num_x$. When k_x= 1, the threshold gate is an OR gate and when $k_x = num_x$, it is an AND gate. Each leaf node x of the tree is described by an attribute and a threshold value $k_x = 1$. To facilitate working with the access trees, we define a few functions. Parent of node x is denoted as parent(x). The function att(x) is denotes the attribute associated with the leaf node x in the tree. In a access tree T the children of a node are numbered from 1 to n(num) . It also defines an ordering between the children of every node. For a node in an access structure, based on the given key the index values are uniquely assigned in an arbitrary manner. The function index(x) returns a number associated with the node x.

B. Cyclic Group

In algebra, a cyclic group is a group that is generated by a single element. Every element can be written as a power of some particular element "g" in multiplicative notation, or .as a multiple of "g" in additive notation. This element "g" is called a "generator" of the group. Any infinite cyclic group is isomorphic to Z, the integers with addition as the group operation. Any finite cyclic group of order n is isomorphic to Z/nZ, the integers modulo n with addition as the group operation [16].

C .Bilinear Paring Algorithm

Pairing-based cryptography is the use of a pairing between elements of two cryptographic groups to a third group to construct cryptographic systems. If the same group is used for the first two groups, the pairing is called symmetric and is a mapping from two elements of one group to an element from a second group [15]. In this way, pairings can be used to reduce a hard problem in one group to a different, usually easier problem in another group.

D. Secured Hashing Algorithm

There are several similarities in the evolution of hash function and that of symmetric block ciphers. We have seen that the increasing power of brute-force attacks[12].Cryptanalysis have led to the decline in the popularity of DES and in the design of newer algorithm with longer key lengths and with features designed to resist specific cryptanalytic attacks. Similarly, advances in computing power and hash function cryptanalysis have led to the decline in the popularity of first MD4 and then MD5, two very popular hash functions. In response, newer hash algorithm have been developed with longer hash code length and with features designed to resist specific cryptanalytic attacks.

IV. PROPOSED WORK

The verifiability of the cloud's transformation and a technique to verify the correctness of the transformation is provided. Initially it modifies the original model of ABE with outsourced decryption then the existing to permit for verifiability of the transformations. Once describing the formal definition of verifiability, we tend to propose a new ABE model and supported this new model construct a concrete ABE theme with verifiable outsourced decryption. Abe scheme with verifiable outsourced decryption and recoverability consists of seven algorithms namely Setup, KeyGen, Encrypt, Decrypt, $GenTk_{Out}$, $Transform_{out}$, and DecryptOut. A trusted Party uses the SetUp algorithmic rule to come up with the general public parameters and a master secret key, and uses KeyGenOut to come up with a non-public key. Encrypt algorithmic rule uses the general public parameters and access structure to cipher the message. In Outsourced Decryption the user uses the $GenTk_{Out}$ algorithmic rule to come up with the transformation key and the retrieving key. The user sends the transformation key to the cloud. Taking as input the transformation key given by a user and a cipher text, the cloud will use the algorithmic rule $Transform_{out}$ to rework the cipher text into a straightforward ciphertext. If the user's attribute satisfies the access structure related to the cipher text; then the user uses the $\text{Decrypt}_{\text{Out}}$ algorithmic rule to recover the plaintext from the transformed cipher text. It takes input as cipher text, public parameters and therefore the transformed cipher text. The hashed blocks of original message are compared with the hashed blocks of retrieved message; if any change within the block then we can confirm that the remaining blocks are original. User splits the original message in to fixed size blocks, and for each block sha1 algorithm is applied. From the resultant 160 bits a 4 bytes can be selected for each block by random hash block function. The resultant random hashed blocks can be stored in the user side. After retrieving the data from the cloud the verification operation is performed. If the verification results the data is modified then to identify the modified block and recover the remaining content Random hash function is applied to the retrieved data. The results are compared with the stored values and the modified blocks can be identified. A hash function H is a transformation that takes a variable-size input m and returns a fixed-size string, that is termed the hash value h (i.e,h = H(m)). Hash functions with this property have a range of general computational uses, however once utilized in cryptography the hash functions are a unit typically chosen to possess some extra properties. Secure Hash Algorithm (SHA) SHA1 is often used on the net to verify the integrity of computer code archives, as a novel symbol, and for digital signatures. The SHA takes a message of but 264 bits long. It supports a style of MD4 with many key variations. It has the following algorithms,

- SetUp
- KeyGen
- Encrypt
- Decrypt
- Verifiable
- Recovery

A. Setup

SetUp (U, λ). The setup algorithm is used to generate the public parameters PK and a master key MK. It takes input as a implicit security parameter.

B. KeyGen

Key Gen (MK,S). The key generation algorithm is used to generate a private key SK to the users by providing the master key MK and a set of attributes that defines the key.

C. Encrypt

Encrypt (PK,M, A). The encryption algorithm uses the private and public keys to encrypt the message for secure transformation. Here a message is chosen along with its public key parameters PK and an access structure among the set of attributes. With these, information provided this algorithm encrypts the given message M and produces a cipher text CT. thus any user who can possess the set of attributes by satisfying the set of attributes can decrypt the message.

D. Decrypt

Decrypt (PK, CT, SK). The decryption algorithm uses both the public and private key to extract the original message sent by the user. Here the receiver receives the encrypted message from the sender. When the message is received, the receiver verifies the set of attributes by using the public key parameters of the sender, the received ciphertext CT with its access policy A, and its private key SK. If it is satisfied then original message m is decrypted using this algorithm.

E. Verifiable

Verifiable (SK, S˜). The delegate algorithm takes a input a secret key SK for some set of attributes S and a set S' ⊆ S. It output a secret key SK for the set of attributes S'.

F. Recovery

The random hashed blocks of original message is compared with the random hashed blocks of retrieved message, if any change in the block the remaining data has been conformed as original message and it is recovered.

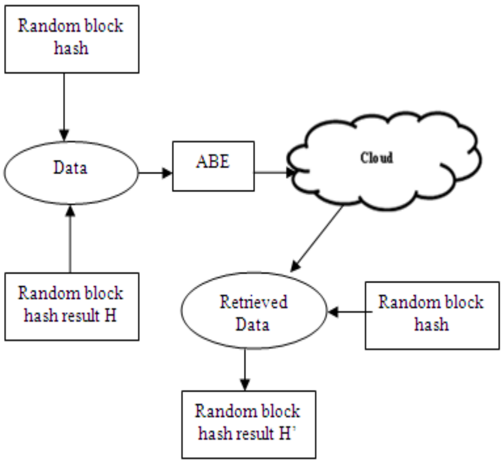

Fig 1.Identifying the modified blocks.

V. METHODOLOGIES USED

A. SetUp()

The setup algorithm takes as input a security parameter λ and a small universe description U={1,2,3…, ℓ}. It first runs G(λ) to obtain (p,G,G_T,e), where G and and G_T are cyclic groups of prime order . It then chooses g,u,v,d $\sum$ G,and α,a $\sum$ Z^*_p uniformly at random,for each attribute i $\sum$ U,It chooses a random value S_i $\sum$ Z^*_p Finally, it chooses a collision-resistant hash function H:G→ Z^*_p .The public parameters PK = (G,G_T ,e,g,u,v,d,g^a , e(g, g) $^{\alpha}$,T_i =g^{si} ∀ i ,H).The master secret key MSK= α.

B. KeyGen()

The key generation algorithm randomly picks t $\sum$ Z^*_p The secret key SK_s = (S,K,$K_{0,}$ K_i) is computed as

$$K = g^{\alpha} \ \ g^{at}$$
$$K_0 = g^t$$
$$K_i = T^t_i \ \forall \ i \in s$$

C. Encrypt ()

The Encrypt algorithm uses the public parameters, Message and Access structure, Access structure consists of attributes and their mapping.

$$C=u^{H(M)} \ v^{H(M)} \ d$$
$$C_1 =M.e(g,g)^{\alpha s}$$
$$C'_{1\,=}\ g^s$$
$$C_{1,i} =g^{a,Ai.v} \ T^{-r1,i} \ \rho(i)$$
$$D_{1,i} =g^{r1,i} \ \forall \ i \in \{1,,2,\ldots,l\}$$
$$C_2 =M.e(g,g)^{\alpha s'}$$
$$C'_{2\,=}\ g^{s'}$$
$$C_{2,i} =g^{a,Ai.v'} \ T^{-r2,i} \ \rho(i)$$
$$D_{2,i} =g^{r2,i} \ \forall \ i \in \{1,,2,\ldots,l\}$$

Encrypted data CT = ((A, ρ),ĉ, C_1 ,C'_1 , $C_{1,i}$, $D_{1,i}$, C_2 ,C'_2 ,$C_{2,i}$, $D_{2,i}$)

D. GenTk$_{Out}$()

In verifiable outsourced decryption the user uses $GenTk_{out}$ algorithm for generating the transformation key "TKs" and Retrieving Key "RK_s ".It takes inputs as the Public parameters and user's secret key"SK_s" The user send the Transformation key to the cloud.

SK_s = (S,K,$K_{0,}$ K_i)

It chooses a random value z $\sum$ Z^*_p

Transformation key TK_s = (S,K',$K'_{0,}$ K'_i)

Retrieving Key RK_s = z

E. Transform$_{out}$()

By using the $transformation_{out}$ algorithm cloud will generate the transformed ciphertext.This algorithm takes as input the public parameters PK ,cipher text CT,and the transformation key TK_s to generate the transformed cipher text CT'.It send the transformed cipher text to the user[20].

$$T'_1 = [e(c'_1 , K')] \ / \ [(\textstyle\prod_{i\varepsilon I} \ (e(C_{1,i} ,K'_0) \ . \ e(K'\rho_{(i)} ,D_{1,i}))^{\omega i})]$$
$$= [e(g,g)^{\alpha s/z} \, e(g,g)^{ats/z} \ ] \ / \ [(\textstyle\prod_{i\varepsilon I} \ e(g,g)^{atAi.v.\omega i/z})]$$
$$= \ e(g,g)^{\alpha s/z}$$
$$T'_2 =[\ e(c'_2 , K') \ ] / [\ (\textstyle\prod_{i\varepsilon I} \ (e(C_{2,i} ,K'_0) \ . \ e(K'\rho_{(i)} ,D_{2,i}))^{\omega i})]$$
$$=[e(g,g)^{\alpha s'/z} \, e(g,g)^{ats'/z} \ ] / [(\textstyle\prod_{i\varepsilon I} \ e(g,g)^{atAi.v'.\omega i/z})]$$
$$= \ e(g,g)^{\alpha s'/z}$$

Transformed Ciphertext ,

CT' = (T = C ,T_1 = C_1 , T'_1 , T'_2 = C_2 , T'_2) .

F. Decrypt$_{Out}$()

Decrypt algorithm uses the public parameters, transformed ciphertext, and cipher text for verification.

PK = (G,G_T ,e,g,u,v,d,g^a ,e(g, g) $^{\alpha}$,T_i =g^{si} ∀ i ,H)

CT = ((A, ρ),ĉ, C_1 ,C'_1 , $C_{1,i}$, $D_{1,i}$, C_2 ,C'_2 , $C_{2,i}$, $D_{2,i}$)

CT' = (T = C ,T_1 = C_1 , T'_1 , T'_2 = C_2 , T'_2) .

RK_s = z

G. Recovery ()

Data is splited into blocks and for each block random hash is applied. Then for each block random hash is applied. The results are stored in the user side.

$$H = \textstyle\sum^{n}_{i=o} h(ct_{=i})$$
$$H = \{h1, h2, \ldots, hn\}$$

After retrieving the data to identify the modified block, random hash function is applied for each block.

$$H' = \textstyle\sum^{n}_{i=o} h(ct_{=i})$$
$$H' = \{h1,h2\ldots.hn\}$$

H and H' values are compared; the blocks whose random hash values are not equal are the modified blocks.

V. PERFORMANCE ANALYSIS

The system ABE-ODR outsources a better performance than the previous systems. The system uses SHA1 hash function where the encryption is done for the blocks separately.

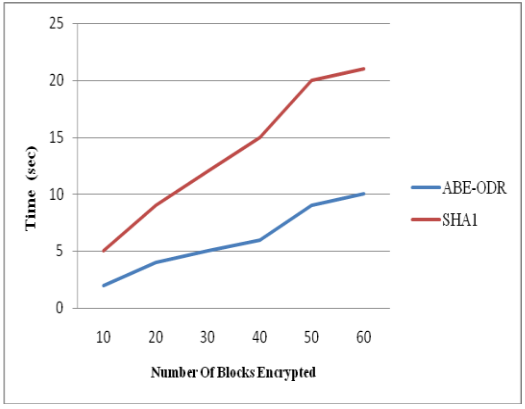

Fig 2.Performance based on Transmission time

Each block consists of 160 bits and each block is encrypted separately and stored in the cloud which requires a huge memory space. In the proposed system ABE-ODR a random hash function is used. Here the bits in the blocks are X-ORed with all the other blocks and finally only a single block of size 160 bits is saved in the cloud. Thus only a small amount of memory is used. Simulation result in Fig 2 shows that the number of blocks encrypted in ABE-ODR is increased when compared to the existing system.

VI. CONCLUSION

In this paper, we tend to think of a brand new demand of ABE with outsourced decryption with recoverability. We tend to change the initial model of ABE with outsourced decryption with verifiability. We planned a concrete ABE theme with verifiable outsourced decryption and proved that it is secure and verifiable and after indentifying the modified data the remaining content can be recovered. Our theme doesn't depend on random oracles. To assess the usefulness of our theme, we implemented it and conducted experiments in an exceedingly simulated outsourcing environment. The theme considerably reduced the computation time needed for resource-limited devices to recover plaintexts.

REFERENCES

[1] L. Cheung and C. C. Newport, "Provably secure ciphertext policy ABE," in Proc. ACM Conf. Computer and Communications Security,2007, pp. 456–465.

[2] V.Goyal, O. Pandey,A. Sahai, and B.Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc. ACM Conf. Computer and Communications Security, 2006, pp. 89–98.

[3] Allison Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters."Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption". In EUROCRYPT, pages 62–91, 2010.

[4] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in Proc. Public Key Cryptography, 2011, pp. 53–70.

[5] N. Attrapadung, J. Herranz, F. Laguillaumie, B. Libert, E. de Panafieu,and C. Ràfols, "Attribute-based encryption schemes with constant-size ciphertexts," Theor. Comput. Sci., vol. 422, pp. 15–38, 2012.

[6] A. Shamir,"Identity-based cryptosystems and signature schemes". In Proceedings of CRYPTO'84 on Advances in Cryptology, pages 47–53, 1985.

[7] M. Chase,"Multi-authority attribute-based encryption". In Proceedings of the 4th IACR Theory of Cryptography Conference (TCC'07), 2007.

[8] T. Okamoto and Uchiyama. "A New Public-Key Cryptosystem as Secure as Factoring". Euro crypt '98, pp. 308–318.

[9] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," in Proc. NDSS, San Diego, CA, USA, 2005.

[10] A. Beimel, "Secure Schemes for Secret Sharing and Key Distribution," Ph.D. dissertation, Israel Inst. of Technology, Technion City, Haifa, Israel,1996.

[11] S. Hohenberger and B. Waters, "Attribute-based encryption with fast decryption," in Proc. Public Key Cryptography, 2013, pp. 162–179.

[12] L.Cheung and C. C. Newport, "Provably secure ciphertext policy ABE," in Proc. ACM Conf. Computer and Communications Security,2007, pp. 456–465.

[13] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute based encryption," in Proc. IEEE Symp. Security and Privacy, 2007, pp. 321–334.

[14] R. Canetti, S. Halevi, and J. Katz."Chosen ciphertext security from identity based encryption". In Advances in Cryptology – Euro crypt, volume 3027 of LNCS, pages 207–222, 2004.

[15] D. Boneh, C. Gentry, and B. Waters."Collusion resistant broadcast encryption with short ciphertexts and private keys". Lecture Notes in Computer Science,3621, 2005. Advances in Crytology – CRYPTO'05.

[16] X.Boyen and B,Waters. "Full-domain subgroup hiding and constant-size group signatures". In Public Key Cryptography, LNCS 4450, pages 1–15. Springer, 2007.

[17] D. Boneh and X. Boyen."Efficient selective-ID secure identity-based encryption without random oracles". In EUROCRYPT, LNCS 3027, pages 223–238, 2004.

[18] M. Bellare, J. A. Garay, and T. Rabin."Fast batch verification for modular exponentiation and digital signatures". In EUROCRYPT, pages 236–250, 1998.

[19] J. Baek and Y. Zheng."Identity-based threshold decryption in Public Key Cryptography", LNCS 2947, pages 262–276. Springer, 2004.

[20] Junzuo Lai, Robert H. Deng, Chaowen Guan, and Jian Weng "Attribute-Based Encryption With Verifiable Outsourced Decryption" in IEEE transactions on information forensics and security, vol. 8, no. 8, August 2013.

AUTHOR'S PROFILE

R.V.Agalya obtained her Bachelor's Degree (2012) in Computer Science and Engineering from Ponjesly college of Engineering under Anna University, Chennai. Now doing her Master Degree (second year) in Computer and Communication from Cape Institute of Technology under Anna University, Chennai.Her research interest is on Cloud Computing.

K. Karthika Lekshmi, obtained her Bachelor's Degree (1997) in Computer Science and Engineering from Bharathiyar University, India. Then she obtained her Master's Degree (2007) in Computer Science and Engineering from Anna University, India. She is pursuing her Ph.D Doctoral Research in Information and Communication Engineering from Anna University, Chennai. Currently, she is an Assistant Professor at the Department of Information Technology in Cape Institute of Technology, Levengipuram affiliated to Anna University, Chennai. Her research interest includes Cloud Computing and Information Security.