Protected Information Recovery For Decentralized Interruption Sympathetic Military Networks

Mr. K.KRANTHI KUMAR ,student of M.Tech,depat.of CSE in Lingayas Institute Of Management And Technology.
Prof. A .Rajesh, Department Of CSE In Lingayas Institute Of Management And Technology

Abstract—A *secure data retrieval scheme using CP-ABE for decentralized DTNs where multiple key authorities manage their attributes independently. We demonstrate how to apply the proposed mechanism to securely and efficiently manage the confidential data distributed in the disruption-tolerant military network. Mobile nodes in military environments such as a battlefield or a hostile region are likely to suffer from intermittent network connectivity and frequent partitions. Disruption-tolerant network (DTN) technologies are becoming successful solutions that allow wireless devices carried by soldiers to communicate with each other and access the confidential information or command reliably by exploiting external storage nodes. Some of the most challenging issues in this scenario are the enforcement of authorization policies and the policies update for secure data retrieval. Ciphertext-policy attribute-based encryption (CP-ABE is a promising cryptographic solution to the access control issues.However, the problem of applying CP-ABE in decentralized DTNs introduces several security and privacy challenges with regard to the attribute revocation, key escrow, and coordination of attributes issued from different authorities.*

Key words*—Access control, attribute-based encryption (ABE), disruption-tolerant network (DTN), multiauthority, secure data retrieval.*

1. INTERDUCTION

Disruption-tolerant network (DTN) technologies are becoming thriving solutions that allow nodes to communicate with each other in these extreme networking environments. Typically, when there is no end-to-end connection between a source and a destination pair, the e-mail from the source node may need to wait in the intermediate nodes for a substantial amount of time until the connection would be eventually established. Roy and Chuah introduced storage nodes in DTNs where data is stored or replicated such that only authorized mobile nodes can access the necessary information quickly and efficiently. Many military applications require increased protection of confidential data including access control methods that are cryptographically enforced.

It is advantageous to make available discriminate access services such that data access policies are defined over user attributes or roles, which are managed by the solution powers that be. For example, in a disruption-tolerant military network, a commander may store confidential in rank at a storage node, which should be accessed by members of "Battalion 1" who are participating in "expanse 2." In this case, it is a practical hypothesis that multiple key authorities are likely to manage their own dynamic attributes for soldiers in their

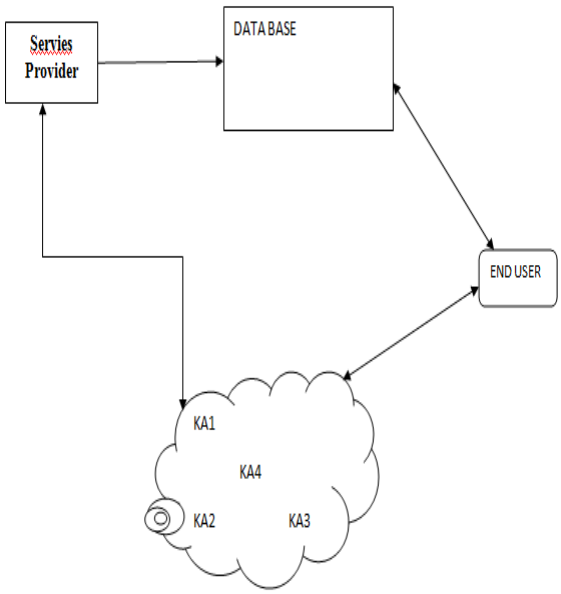
deployed regions or echelons, which could be recurrently changed (e.g., the attribute representing

current location of moving soldiers). We refer to this DTN architecture where multiple authorities issue and manage their own attribute keys independently as a decentralized DTN.

The concept of attribute-based encryption (ABE) is a promising approach thatfulfills the requirements for secure data retrieval in DTNs. ABE features a mechanism that enables an access control over encrypted data using access policies and ascribed attributes among private keys and ciphertexts. Especially, ciphertext-policy ABE (CP-ABE) provides a scalable way of encrypting data such that the encryptor defines the attribute set that the decryptor needs to possess in order to decrypt the ciphertext. Thus, different users are allowed to decrypt different pieces of data per the security policy. However, the problem of applying the ABE to DTNs introduces several security and privacy challenges. Since some users may change their associated attributes at some point (for example,

moving their region), or some private keys might be compromised, key revocation (or update) for each attribute is necessary in order to make systems secure. However, this issue is even more difficult, especially in ABE systems, since each attribute is conceivably shared by multiple users (henceforth, we refer to such a collection of users as an attribute group). This implies that revocation of any attribute or any single user in an attribute group would affect the other users in the group. For example, if a user joins or leaves an attribute group, the associated attribute key should be changed and redistributed to all the other members in the same group for backward or forward secrecy. It may result in bottleneck during rekeying procedure or security degradation due to the windows of vulnerability if the previous attribute key is not updated immediately.

Another challenge is the key escrow problem. In CP-ABE, the key authority generates private keys of users by applying the authority's master secret keys to users' associated set of attributes. Thus, the key authority can decrypt every ciphertext addressed to specific users by generating their attribute keys.



1.1. Figure Military Networks

The last challenge is the coordination of attributes issued from different authorities. When multiple authorities manage and issue attribute keys to users independently with their own master secrets, it is very hard to define fine-grained access policies over attributes issued from different authorities. For example, suppose that attributes "role 1" and "region 1" are managed by the authority A, and "role 2" and "region 2" are managed by the authority B. Then, it is impossible to generate an access policy (("role 1" OR "role 2") AND ("region 1" or "region 2")) in the previous schemes because the OR logic between attributes issued from different authorities cannot be implemented. This is due to the fact that the different authorities generate their own attribute keys using their own independent and individual master secret keys. Therefore, general access policies, such as "-out-of-" logic, cannot be expressed in the previous schemes, which is a very practical and commonly required access policy logic.

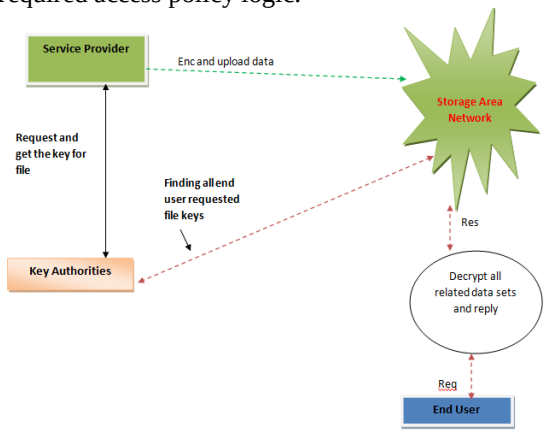


Fig .block diagram

2. LINKED EFFORT

The solution clout chooses a policy for each user that determines which cipher text he can decrypt and issues the key to each user by embed the policy into the user's key. nevertheless, the roles of the cipher texts and keys are reversed in CP-ABE. In CP-ABE, the cipher text is encrypted with an access policy chosen by an encrypted, but a key is simply created with respect to an attributes set. CP-ABE is more appropriate to DTNs than KP-ABE because it enables encryptions such as a commander to choose an access policy on attributes and to encrypt confidential data under the access structure via encrypting with the corresponding public keys or attributes.

2.1 feature Revocation:

Bettencourt *et al.* and Boldyreva *et al.* first suggested key revocation mechanisms in CP-ABE and KP-ABE, respectively. Their solutions are to append to each attribute an expiration date (or time) and distribute a new set of keys to valid users after the expiration. The periodic attribute revocable ABE schemes have two main problems.

1.It is a large set-up that users such as defense force may change their attributes frequently, e.g., position

or location move when taking into consideration these as attributes. Then, a user who newly holds the attribute might be able to access the previous data encrypted before he obtains the attribute until the data is encrypted with the newly updated attribute keys by periodic rekeying (backward confidentiality). For example, assume that at time, a cipher text is encrypted with a policy that can be decrypted through a set of attributes (embedded in the users keys) for client with. After time, say, a user newly holds the point set. Even if the new user should be disallowed on the way to decrypt the cipher text for the time instance, he can still decrypt the previous cipher text until it is re encrypted with the newly updated attribute keys. he can still decrypt the cipher text of the previous time instance unless the key of the user is expired and the cipher text is encrypted with the newly updated key that the user cannot obtain. We call this uncontrolled period of time windows of vulnerability.

The other is the scalability problem. The key authority periodically announces a key update material by unicast at each time-slot so that all of the nonrevoked users can update their keys. This results in the "1-affects- " problem, which means that the update of a single attribute affects hewhole nonrevoked users who share the attribute. This could be a bottleneck for both the key authority and all nonrevoked users. The immediate key revocation can be done by revoking users using ABE that supports negative clauses. To do so, one just adds conjunctively the AND of negation of revoked user identities (where each is considered as an attribute here).

The size of private key over the original CP-ABE scheme of Bettencourt *et al.*, where is the maximum size of revoked attributes set. Galle *et al.* also proposed a user revocable KP-ABE scheme, but their scheme only works when the number of attributes associated with a cipher text is exactly half of the universe size.

2.4 **Key Escrow**

Most of the existing ABE schemes are constructed on the architecture where a single trusted authority has the power to generate the whole private keys of users with its master secret information. Thus, the key escrow problem is inherent such that the key authority can decrypt every cipher text addressed to users in the system by generating their secret keys at any time. Chase *et al.* presented a distributed KP-ABE scheme that solves the key escrow problem in a multiauthority system. In this approach, all (disjoint) attribute authorities are participating in the key generation protocol in a distributed way such that they cannot pool their data and link multiple attribute sets belonging to the same user. One disadvantage of this fully distributed approach is the performance degradation. Since there is no centralized authority with master secret information, all attribute authorities should communicate with each other in the system to generate a user's secret key. This results in communication overhead on the system setup and the rekeying phases and requires each user to store additional auxiliary key components besides the attributes keys, where is the number of authorities in the system.

2.5 Decentralized ABE:

Proposed decentralized CP-ABE schemes in the multiauthority network environment. For example, let be the key authorities, and be attributes sets they independently manage, respectively. Then, the only access policy expressed with is , which can be achieved by encrypting a message with by , and then encrypting the resulting cipher text with by (where is the cipher text encrypted under), and then encrypting resulting cipher text with by , and so on, until this multiencryption generates the final cipher text . Thus, the access logic should be only AND, and they require iterative encryption operations where is the number of attribute authorities. Therefore, they are somewhat restricted in terms of expressiveness of the access policy and require computation and storage costs. Chase and Elko *et al.* proposed multiauthority KP-ABE and CP-ABE schemes, respectively. However, their schemes also suffer from the key escrow problem like the prior decentralized schemes.

2.6 Involvement

In this organization, we propose an attribute-based secure data repossession scheme using CP-ABE for decentralized DTNs. The proposed scheme features the following achievements. First, immediate attribute revocation enhances backward/forward secrecy of private data by reducing the windows of vulnerability. Second, encryptors can define a fine-grained access policy using any monotone right to use structure under attributes issued from any chosen authorities.

3. PRELIMINARIES AND DEFINITION

The Trusted Platform Module (TPM) is a microcontroller that conforms to the specification established by the Trusted Computing Group (TCG)[1]. The TCG website states, "The TPM is a microcontroller that stores keys, passwords and digital certificates." The TPM is at the heart of the Trusted Computing (TC) initiative, as it provides the root of trust as well as capabilities for many TC

.

applications. The TPM is usually attached to a PC motherboard but could potentially be used in any computing device that requires TC capabilities.

In a few words, the TPM provides a safe place to store sensitive information, provides a protected space for key operations and other security critical tasks, and stores and reports integrity measurements. It is specifically designed to enhance platform security beyond the capabilities of software and shield keys and other sensitive information from software-based attacks. The TPM is intended to complement existing specifications, such as X.509, IPSEC, VPN, PKI, S/MIME, and SSL.

3.1 Sensitive Information

The TPM and other elements of the TCG specifications are designed to protect against or mitigate the potential damage caused by a variety of threats and attacks. This paper focuses on those that affect PC clients (desktops and notebooks). PC clients have a large number of vulnerabilities, known and unknown, and this is unlikely to change given the nature and practices of the software industry. In addition, keeping patches up to date for all software installed on a system is time consuming and a large percentage of systems do not have all applicable patches. While networks and servers offer the most value for attackers, they are also better protected than PC clients. In addition, PC clients often contain information, such as keys and passwords that can be used to access and compromise networks and servers or can be used for distributed attacks, such as Distributed Denial of Service (DDoS), against them. Keys could also be used to decrypt sensitive information, steal a digital identity, or forge signatures. PC clients also contain information, such as credit card and social security numbers, that is itself valuable. As a result of these and other factors, attackers are increasingly focused on PC clients. TPMs should support preventing attackers from being able to find information on a compromised client that can be used to compromise another system for which the client or its user has access. The TPM should also enable a network administrator to prevent a compromised client from being able to compromise or disrupt the rest of the network.

The information on clients could include encryption or signing keys, passwords, and personal or proprietary information. The TPM is designed to protect sensitive information on PC clients as well as the servers and networks they may connect to. In addition, some private RSA keys never leave the TPM, so it is impossible to obtain them directly by software means. The TPM does *not* attempt to reduce the number of vulnerabilities in software or prevent an attacker from exploiting those vulnerabilities. Instead, the TPM seeks to detect when the client is compromised and limit the damage and protect sensitive information when it occurs. If the TPM and related software are configured correctly, the attacker cannot access the sensitive information regardless of what he or she does. Attacks on sensitive information should be no better than a brute force attack.

One primary attack that the TPM seeks to thwart is attack on keys when cryptographic operations are performed in software. It has been definitively proven[2] that even very good encryption is vulnerable to attack performed in the usual locations, such as memory. TPM cryptography operations are performed in a closed hardware environment, protecting the keys at their most vulnerable point.

The TPM should prevent theft (copying to another system for use there) of RSA keys as well as improper use of keys when the system has been compromised. The latter is very dependent on the system firmware (i.e. BIOS), TPM Software Stack (TSS) and how they work detect that a system has been compromised, but the TPM provides all necessary framework.

The TPM also allows multiple users to protect sensitive information on a shared client. Even if a user has permission to use the client, they still may not have access to other user's secrets.

If any encryption key-pair is compromised, the data it protects and any data protected by keys that it protects may also be compromised. Once an encryption key-pair is compromised, all data ever encrypted with it is compromised and this cannot be recovered from, except by deleting all copies of the data encrypted with that key (including ones that may have been stolen). Likewise, once a digital signature key is compromised, the attacker can sign anything they wish. If certificates are used, the certificate could be revoked. The TPM cannot detect compromises of its own keys. Instead it protects them by not letting some private keys leave the TPM, encrypting its keys when they leave the TPM, and detecting compromise of the client software.

Cryptographic Mechanisms and Algorithms

These and other cryptographic features are described in the following sections.

- Indiscriminate Quantity Making (IQM)
- Asymmetric key (RSA) and nonce generation
- Asymmetric encryption/decryption (RSA)
- Signing (RSA)
- Hashing (SHA-1)
- Keyed-Hash Message Authentication Code (HMAC)

The specification allows TPMs to implement additional features or algorithms, such as DSA or elliptic curve asymmetric algorithms, but "there is no guarantee that these keys can migrate to other TPM devices or that other TPM devices will accept signatures from these additional algorithms." The TPM specification stipulates minimum key lengths for some uses. Storage keys, for example, must be equivalent in strength to a 2048-bit or greater RSA key.

4. FUTURE PROPOSAL

I provide a multiauthority CP-ABE scheme for secure data retrieval in decentralized DTNs. Each local authority issues partial modified and attribute key components to a user by performing secure 2PC protocol with the central authority. Each attribute key of a user can be updated separately and immediately. Thus, the scalability and security can be enhanced in the proposed scheme. The Trusted Platform Module is the root of trust and a central component for Trusted Computing. It includes several types of cryptographic capabilities, including RSA encryption and digital signing, SHA-1 hashing, HMACs, and a random number generator. It also provides hardware protection for these capabilities and sensitive information on the client. In addition, the TPM provides platform authentication and attestation features. The purpose of the TPM is not to prevent attacks on clients. Instead, its focus is on detecting when a client has been compromised and protecting sensitive information, the network, and other systems. Along with software, the TPM features help protect users, their sensitive information, and the infrastructure in the presence of software vulnerabilities.Since the first CP-ABE scheme proposed by Bettencourt *et al.* dozens of CP-ABE schemes have been proposed. The subsequent CP-ABE schemes are mostly motivated by more rigorous security proof in the standard model. However, most of the schemes failed to achieve the expressiveness of the Bethencourt *et al.*'s scheme, which described an efficient system that was expressive in that it allowed an encryptor to express an access predicate in terms of any monotonic formula over attributes. Therefore, in this section, we develop a variation of the CP-ABE algorithm partially based on (but not limited to) Bethencourt *et al.*'s construction in order to enhance the expressiveness of the access control policy instead of building a new CP-ABE scheme from scratch.

1) CA first selects a random r', and sends $g^{r_t - r}$ and g^r to A_i and u_t, respectively.
2) A_i takes a set of attributes $\Lambda_i \subseteq A_i(\mathcal{L})$ as inputs and outputs a set of attribute keys for the user that identifies with that set Λ_i. It chooses random $r_j \in \mathbb{Z}_p^*$ for each attribute $\lambda_j \in \Lambda_i$. Then, it gives the following secret value to the user u_t:

$$\forall \lambda_j \in \Lambda_i \ : \ D_j = g^{r_t - r'} \cdot H(\lambda_j)^{r_j}, D'_j = g^{r_j}.$$

Then, the user computes $g^{r'} \cdot D_j$ for all its attributes key components and finally obtains its whole secret key set as

$$\mathrm{SK}_{u_t} = \left(D = g^{\frac{(\alpha_1 + \cdots + \alpha_m) + r_t}{\beta}}, \forall \lambda_j \in S : D_j = g^{r_t} \cdot H(\lambda_j)^{r_j}, D'_j = g^{r_j}\right)$$

where $S = \bigcup_{i=1}^{m} \Lambda_i$.

Data encryption

The encryption algorithm chooses a polynomial Qx

Let Y be the set of leaf nodes in the access tree. To encrypt a message $M \in \mathbb{G}_1$ under the tree access structure $\mathcal{T}$, it constructs a ciphertext using public keys of each authority as

$$\mathrm{CT} = (\mathcal{T}, \tilde{C} = Me(g,g)^{(\alpha_1 + \cdots + \alpha_m)s}, C = h^s, \forall y \in Y : C_y = g^{q_y(0)}, C'_y = H(\lambda_y)^{q_y(0)}),$$

where $\tilde{C}$ can be computed as $\tilde{C} = M \cdot (\mathrm{PK}_{A_1} \times \cdots \times \mathrm{PK}_{A_m})^s = Me(g,g)^{(\alpha_1 + \cdots + \alpha_m)s}$.

Data decryption

$$F_x = \prod_{z \in S_x} F_z^{\Delta_{i,S'_x}(0)}, \quad \text{where} \quad \begin{matrix} i = \mathrm{index}(z), \\ S'_x = \{\mathrm{index}(z) : z \in S_x\} \end{matrix}$$

$$= \prod_{z \in S_x} \left(e(g,g)^{r_t \cdot q_z(0)}\right)^{\Delta_{i,S'_x}(0)}$$

$$= \prod_{z \in S_x} \left(e(g,g)^{r_t \cdot q_{p(z)}(\mathrm{index}(z))}\right)^{\Delta_{i,S'_x}(0)}$$

$$= \prod_{z \in S_x} e(g,g)^{r_t \cdot q_x(i) \cdot \Delta_{i,S'_x}(0)}$$

$$= e(g,g)^{r_t \cdot q_x(0)}$$

5. MODULES

1. Key Authorities
2. Storage Nodes
3. Sender
4. User

5.1 MODULES DESCRIPTION:

Key Authorities:

They are key generation centers that generate public/secret parameters for CP-ABE. The key authorities consist of a central authority and multiple local authorities. We assume that there are secure and reliable communication channels between a central authority and each local authority during the initial key setup and generation phase. Each local authority manages different attributes and issues corresponding attribute keys to users. They grant differential access rights to individual users based on the users' attributes. The key authorities are assumed to be honest-but-curious. That is, they will honestly execute the assigned tasks in the system; however they would like to learn information of encrypted contents as much as possible.

Storage node:

This is an entity that stores data from senders and provide corresponding access to users. It may be mobile or static. Similar to the previous schemes, we also assume the storage node to be semi-trusted that is honest-but-curious.

Sender:

This is an entity who owns confidential messages or data (e.g., a commander) and wishes to store them into the external data storage node for ease of sharing or for reliable delivery to users in the extreme networking environments. A sender is responsible for defining (attribute based) access policy and enforcing it on its own data by encrypting the data under the policy before storing it to the storage node.

User:

This is a mobile node who wants to access the data stored at the storage node (e.g., a soldier). If a user possesses a set of attributes satisfying the access policy of the encrypted data defined by the sender, and is not revoked in any of the attributes, then he will be able to decrypt the ciphertext and obtain the data.

CP-ABE Method:

In Cipher text Policy Attribute based Encryption scheme, the encryption can fix the policy, who can decrypt the encrypted message. The policy can be formed with the help of attributes. In CP-ABE, access policy is sent along with the cipher text. We propose a method in which the access policy need not be sent along with the cipher text, by which we are able to preserve the privacy of the encryption. This techniques encrypted data can be kept confidential even if the storage server is untrusted; moreover, our methods are secure against collusion attacks. Previous Attribute- Based Encryption systems used attributes to describe the encrypted data and built policies into user's keys; while in our system attributes are used to describe a user's credentials, and a party encrypting data determines a policy for who can decrypt.

6. PROPOSED SYSTEM:

We provide a multi authority CP-ABE scheme for secure data retrieval in decentralized DTNs.Each local authority issues partial personalized and attribute key components to a user by performing secure 2PC protocol with the central authority. Each attribute key of a user can be updated individually and immediately. Thus, the scalability and security can be enhanced in the proposed scheme.

Since the first CP-ABE scheme proposed by Bettencourt *et al* , dozens of CP-ABE schemes have been proposed. The subsequent CP-ABE schemes are mostly motivated by more rigorous security proof in the standard model. However, most of the schemes failed to achieve the expressiveness of the Bettencourt *et al.*'s scheme, which described an efficient system that was expressive in that it allowed an encrypted to express an access predicate in terms of any monotonic formula over attributes

ADVANTAGES OF PROPOSED SYSTEM

Data confidentiality: Unauthorized users who do not have enough credentials satisfying the access policy should be deterred from accessing the plain data in the storage node. In addition, unauthorized access from the storage node or key authorities should be also prevented.

Collusion-resistance: If multiple users collude, they may be able to decrypt a cipher text by combining their attributes even if each of the users cannot decrypt the cipher text alone.

Backward and forward Secrecy: In the context of ABE, backward secrecy means that any user who comes to hold an attribute (that satisfies the access policy) should be prevented from accessing the plaintext of the previous data exchanged before he holds the attribute. On the other hand, forward secrecy means that any user who drops an attribute should be prevented from accessing the plaintext of the subsequent data exchanged after he drops the

attribute, unless the other valid attributes that he is holding satisfy the access policy.

7. EXPECTED OUTCOMES

Before implementing my project the below expected outputs are list below:

1. Register into sender details and login user details.
2. Run as the Router and keyauthority1, keyauthority2, keyauthority3.
3. After entered into details run as the sender and uploading a file.
4. The file will send to the keyauthorities and returned to the sender
5. After uploading a file into got a successfully uploaded message.
6. After got a successfully uploaded message data is encrypted.
7. Then the file is encrypted and uploaded data got a secret key.
8. After getting the secret key,run the user who had login
9. By using the secrete key user receives the file.
10. File is successfully received

8. CONCLUSION

DTN technologies are becoming successful solutions in military applications that allow wireless devices to communicate with each other and access the confidential information reliably by exploiting external storage nodes. CP-ABE is a scalable cryptographic solution to the access control and secure data retrieval issues. In this paper, we proposed an efficient and secure data retrieval method using CP-ABE for decentralized DTNs where multiple key authorities manage their attributes independently. The inherent key escrow problem is resolved such that the confidentiality of the stored data is guaranteed even under the hostile environment where key authorities might be compromised or not fully trusted. In addition, the fine-grained key revocation can be done for each attribute group. We demonstrate how to apply the proposed mechanism to securely and efficiently manage the confidential data distributed in the disruption- tolerant military network.

9.REFERRENCE

[1] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxpop: Routing for vehicle-based disruption tolerant networks," in Proc. IEEE INFOCOM, 2006, pp. 1–11.

[2] M. Chuah and P. Yang, "Node density-based adaptive routing scheme for disruption tolerant networks," in Proc. IEEE MILCOM, 2006, pp. 1–6.

[3] M. M. B. Tariq, M. Amar, and E. Zequra, "Message ferry route design for sparse ad hoc networks with mobile nodes," in Proc. ACM MobiHoc, 2006, pp. 37–48.

[4] S. Roy and M. Chuah, "Secure data retrieval based on cipher text policy attribute-based encryption (CP-ABE) system for the DTNs," Lehigh CSE Tech. Rep., 2009.

[5] M. Chuah and P. Yang, "Performance evaluation of content-based information retrieval schemes for DTNs," in Proc. IEEE MILCOM, 2007, pp. 1–7.

[6] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, "Plutus: Scalable secure file sharing on untrusted storage," in Proc. Conf. File Storage Technol., 2003, pp. 29–42.

[7] L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, "Mediated ciphertext-policy attribute-based encryption and its application," in Proc. WISA, 2009, LNCS 5932, pp. 309–323.

[8] N. Chen, M. Gerla, D. Huang, and X. Hong, "Secure, selective group broadcast in vehicular networks using dynamic attribute based encryption," in Proc. Ad Hoc Netw. Workshop, 2010, pp. 1–8.

[9] D. Huang and M. Verma, "ASPE: Attribute-based secure policy enforcement in vehicular ad hoc networks," Ad Hoc Netw., vol. 7, no. 8, pp. 1526–1535, 2009.

[10] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," Cryptology ePrint Archive: Rep. 2010/351, 2010.

[11] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in Proc. Eurocrypt, 2005, pp. 457–473.

[12] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc. ACM Conf. Comput. Commun. Security, 2006, pp. 89–98.
[13] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attributebased encryption," in Proc. IEEE Symp. Security Privacy, 2007, pp. 321–334.
[14] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in Proc. ACM Conf. Comput. Commun. Security, 2007, pp. 195–203.
[15] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in Proc. ASIACCS, 2010, pp. 261–270.
[16] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in Proc. ACM Conf. Comput. Commun. Security, 2008, pp. 417–426.
[17] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, "Secure attributebased systems," in Proc. ACMConf. Comput. Commun. Security, 2006, pp. 99–112.
[18] S. Rafaeli and D. Hutchison, "A survey of key management for secure group communication," Comput. Surv., vol. 35, no. 3, pp. 309–329, 2003.
[19] S. Mittra, "Iolus: A framework for scalable secure multicasting," in Proc. ACM SIGCOMM, 1997, pp. 277–288.
[20] P. Golle, J. Staddon, M. Gagne, and P. Rasmussen, "A content-driven access control system," in Proc. Symp. Identity Trust Internet, 2008, pp. 26–35.
[21] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in Proc. ACM Conf. Comput. Commun. Security, 2007, pp. 456–465.
[22] V.Goyal, A. Jain,O. Pandey, andA. Sahai, "Bounded ciphertext policy attribute-based encryption," in Proc. ICALP, 2008, pp. 579–591.
[23] X. Liang, Z. Cao, H. Lin, and D. Xing, "Provably secure and efficient bounded ciphertext policy attribute based encryption," in Proc. ASIACCS, 2009, pp. 343–352.
[24] M. Chase and S. S. M. Chow, "Improving privacy and security inmultiauthority attribute-based encryption," in Proc. ACM Conf. Comput. Commun. Security, 2009, pp. 121–130.
[25] M. Chase, "Multi-authority attribute based encryption," in Proc. TCC, 2007, LNCS 4329, pp. 515–534.
[26] S. S.M. Chow, "Removing escrow from identity-based encryption," in Proc. PKC, 2009, LNCS 5443, pp. 256–276.

BIBLIOGRAPHY

I completed B.Tech in sri sarathi institute of engineering and technology. Now I am studying M.Tech in lingayas institute of management and technology in the year 2013-2015.