INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
*WINGS TO YOUR THOUGHTS.....*

# Secure Attribute Based Disruption-Tolerant Military Networks

**Ms. Selva Sangeetha S[1], Ms. Sharmila S [2] and Ms. Kalaivani M [3]**

[1,2]UG scholar, Department of CSE, Dhanalakshmi College of Engineering, Chennai
[1]selvasangeethas1810@gmail.com [2]sharmi2.sa@gmail.com
[3]Assistant Professor, Department of CSE, Dhanalakshmi College of Engineering, Chennai
[3]dcekalaivani27@gmail.com

***Abstract*— *In many military and remote system situations, associations of remote gadgets conveyed by soldiers may be temporarily separated by sticking, ecological variables, and portability, particularly when they work in antagonistic environments. In the existing system Disruption-Tolerant Network (DTN) technologies allows nodes to communicate with each other in these extreme networking environments by implementing Ciphertext-policy attribute-based encryption (CP-ABE) algorithm which lacks in privacy and security. In this paper, we propose a secure attribute based data retrieval scheme using Rivest Shamir Adleman (RSA) algorithm. When there is no peer-to-peer connection between a sender and a receiver pair, the messages from the sender node may need to wait in the intermediate node for a substantial amount of time until the connection would be eventually established. We introduce usage of Hash based Message Authentication Code (HMAC) in storage nodes to demonstrate how to apply the proposed mechanism such that only authorized mobile nodes can securely and efficiently manage the confidential data in the disruption-tolerant military network.***

***Keywords****- Access control, Ciphertext-policy attribute-based encryption (CP-ABE), Disruption-tolerant network (DTN), multiauthority, secure data retrieval, Hash based Message Authentication Code (HMAC).*

## 1. INTRODUCTION

A Disruption-Tolerant Network is a network which is destined to reduce the impact on temporary communication problems, and drawbacks. DTN is used in fault-tolerant methods and technologies, in the quality controlled decadence under adverse conditions or extreme traffic loads.

In many military network circumstances, connections of wireless devices carried by soldiers may be momentarily disconnected by mobbing, environmental factors, and mobility, especially when they operate in unsociable environments. [1]–[3].Typically, when there is no peer-to-peer connection between a sender and a receiver pair, the messages from the sender node may need to wait in the in-between nodes for a considerable amount of time until the connection would be finally established. Roy [4] and Chuah [5] introduced storage nodes in DTNs where data is stored or not ignored such that only authorized mobile nodes can access the essential information quickly and efficiently.

Many military applications require added on protection of confidential data including access control methods that are cryptographically enforced [6], [7]. In many cases, it is preferable to provide distinguished access services such that data access policies are defined over user attributes, which are managed by the key powers. For instance, in a disturbance tolerant military network, a commander may store high confidential information at capacity hub, which ought to be gotten to by individuals from "District1" who are immersed in "Region 2."

In this case, it is a reasonable supposition that multiple key authorities are likely to manage their own vigorous attributes for soldiers in their deployed regions or at social hierarchy, which could be frequently changed (e.g., the attribute representing current location of moving soldiers) [4], [8], [9]. We refer to this architecture where multiple key authorities issue and manage their own key attributes independently as a decentralized DTN [10], which depends upon unicasting, multicasting, and broadcasting. Unicasting refers to sending of messages from one to another, multicasting refers to sending of messages from one end to multiple receivers, and broadcasting refers to broadcast the messages in public.

Ciphertext-policy Attribute Based Encryption (CP-ABE) provides a flexible way of encrypting data such that the sender who encrypts messages defines the attribute set that the receiver who decrypts messages needs to retain in order to decrypt the ciphertext [11].

Thus, we introduced using of Rivest Shamir Adleman (RSA) algorithm where every member of a group will be provided separate keys for encryption and decryption where different users are allowed to decrypt different pieces of message as per their security policy.

However, the problem of applying the CP-ABE to DTNs introduces several security and privacy challenges, due to same public key and private key for encryption and decryption of data in a group. Since some users may change their associated attributes like their position from one region to another at some point, geo-casting based sending of messages doesn't provide complete security while retrieval of data, or some private keys might be impaired so key updating for each attribute is necessary in order to make systems secure.

However, this issue is difficult, especially in CP-ABE systems, since each attribute is feasibly shared by multiple users of network due to same encryption and decryption keys provided. Previously, key for

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

revocation will be done for regions when a member of a region leaves and gets added in another region, so we introduced an feature of revocating group secret key in particular time and also key revocation will be performed if the node signature at the time of data transfer is not matched with storage head signature.

### 1.1 Related Work

#### 1.1.1 Attribute Revocation

Bethencourt et al. [11] and Boldyreva et al. [12] first suggested key revocation mechanisms in CP-ABE. Their solutions are to append to each attribute an expiration date or time and scatter a new set of keys to valid members after the expiration.

#### 1.1.2 Decentralized ABE

Huang *et al.* [9] and Roy *et al.* [4] created decentralized CP-ABE schemes in the multiauthority network environment. They achieved a combined access policy over the attributes issued from different authorities by simply encrypting data multiple times.

### 1.2 Problem Analysis

Previously, updating of key will not be proceeded among the regions where a member of a region leaves and gets added in another region, so in this paper, we proposed revocating group key for the regions and also to inform all the members of both regions after key updation. Central Authority sends information to a member based on location which fails if a member moves to other location, So in this paper, we proposed features based on unicasting, multicasting, and broadcasting which delivers messages based on mobility of a node.

## 2. NETWORK ARCHITECTURE

In this section we describe the DTN architecture and define security model.

### 2.1 System Description

Figure 1 shows the building design of the DTN. As indicated in Fig. 1, the construction model comprises of the following system entities.

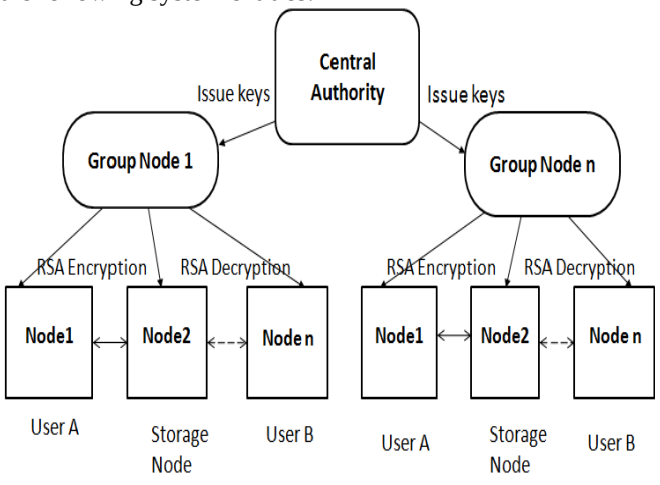


**Figure 1:** Architecture of secure attribute based data retrieval

1) *Central Authority:* They are key generation centres that generate public-private parameters. Central authority is the key authorities. There are secure and reliable communication channels between a central authority and each node during the initial key setup and key generation phase. Central authority manages different attributes and issues corresponding attribute keys to each nodes of their region.
2) *Storage Node:* This is an entity that stores encrypted data from sender and provide corresponding access to users. In proposed scheme, there is no specific node as Storage node. The node for which the Mobility and Power Management is high and efficient acts temporarily as Storage Node.
3) *Sender:* This is an entity who owns encrypted messages or data and stores them into the external data storage node for ease of sharing or for reliable delivery of encrypted message to receiver.
4) *Receiver:* This is a mobile node who wants to retrieve the data stored at the storage node. If a user possesses a set of attributes satisfying the secure access policy and if the HMAC is same for the encrypted data defined by the sender, then he will be able to decrypt the cipher text and obtain the data.

## 3. BASIC DEFINITIONS

### 3.1 Definitions

Let $T$ be a tree which represents an access structure. Every non leaf hub of the tree speaks to limit entryway. On the off chance that numx is the quantity of youngsters hub of a node $x$ and $k_x$ is its threshold value, then, it will be $0 \leq k_x \leq num_x$. Each leaf node of the tree $T$ is described by an attribute and a threshold value $k_x=1$. $\lambda_x$, denotes the key attribute associated with the leaf node in the tree. $\rho(x)$ represents the parent node of the node in the tree $T$.

The children of every node are numbered from 1 to num. The capacity index(x) returns such a number associated with the node. The index values are not commonly assigned to nodes in the access structure for a given key in an uninformed manner.

Let $T_x$ be the subtree of tree $T$ rooted at the node $x$. If a set of attributes $\gamma$ satisfies the access tree $T_x$, we denote it as $T_x(\gamma)=1$. $T_x(\gamma)$ has been recursively computed as shown below. If $x$ is a nonleaf node of a tree, evaluate $T_x'(\gamma)$ for all children $x'$ of node $x$ then $T_x(\gamma)$ returns 1 if and only if at least $k_x$ children return 1, if $x$ is a leaf node, then $T_x(\gamma)$ returns 1 if and only if $\lambda_{x \epsilon} \gamma$.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

## 4. PROPOSED SCHEME

In this section, we used a one of the most practicable public-key cryptosystems RSA Algorithm for secure data transmission. RSA Algorithm is divided into Symmetric and Asymmetric encryption algorithm. In this paper, we use Asymmetric encryption algorithm of RSA that is different from symmetric encryption algorithm. It needs two keys, one is public key another is private/secret key. They appear in pairs, so that the public key to encrypt data, only with its corresponding private key can decrypt the data; Users can do the encryption and decryption using two different keys, so this algorithm is called RSA Asymmetric Encryption Algorithm. It is a kind of algorithm that can be used for not only for data encryption but also for digital signature.

MAC, message authentication codes are used between two parties who share a secret key in order to validate information transferred among these parties, The sender using HASH based Message Authentication Code (HMAC) algorithm sends messages by calculating the hash value of the file M1, then generate the digital signature C1 from using the key to encipher digital abstract and then M1 C1 are made together and sent to the receiver end. The receiver receives the file FM1 and digital signature FC1, needs to verify that M1 and FM1 are same or not.

The verification process is to get hash value H1 of FM1 using the HASH function and get the H2 from the decrypt digital signature of FC1 by using the public key. Compare H1 to H2 that whether they are same or not, if they are same then Information transmission from one node to another node is secure and private.

### 4.1 Algorithm – Rivest Shamir Adleman (RSA)

*1) Key generation:* The keys for the RSA algorithm are generated the following way choose two distinct prime numbers $p$ and $q$. For security purposes, integers $p$ and $q$ should be chosen at random, and should be of similar and long range bit-length. Prime numbers can be efficiently found using a test called primarily test. Compute $n = pq$. $n$ is used as the modulus for both the public and private keys. Its length, usually are expressed in bits, which is the key length. Compute $\varphi(n) = \varphi(p)\varphi(q) = (p-1)(q-1) = n - (p+q-1)$, where $\varphi$ is Euler's totient function.

Choose an integer $e$ such that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$; where $e$ and $\varphi(n)$ are coprime. $e$ is released as the public key example. Focus d as $d \equiv e^{-1} \pmod{\varphi(n)}$; where $d$ is the multiplicative inverse of $e$ (modulo $\varphi(n)$). This is all the more obviously expressed as: $d$ given $d \cdot e \equiv 1 \pmod{\varphi(n)}$ , and this is processed utilizing the extended Euclidean algorithm.

Using the pseudocode in the whole numbers area, inputs an and n relate to e and φ(n), individually, $d$ is key type. The general population key comprises of the modulus n and exponent $e$ and the *private key* consists of the modulus $n$ and exponent $d$, must be kept secret. $p$, $q$, and $\varphi(n)$ must also be kept secret because they can be used by others to calculate $d$.

*2) Data Encryption:* We explain the process of encryption with an example; Sender transmits his/her public key $(n, e)$ to Receiver and keeps the private key $d$ secret. Receiver then wishes to send message $M$ to Sender. Receiver first turns message $M$ into an integer $m$, such that $0 \le m < n$ by using an agreed-upon reversible protocol known as cushioning plan. He then processes the figure content c relating to $c \equiv m^e \pmod{n}$ this can be effectively, actually for 500-bit numbers, using Modular exponentiation. Receiver then transmits $c$ to Sender.

*3) Data Decryption:* Here, Sender can recover $m$ from $c$ by using his/her private key exponent $d$ by computing $m \equiv c^d \pmod{n}$ Given $m$, Sender can recover the original message $M$ by reversing the padding scheme.

Here, suppose sender wishes to send a signed message to receiver, he/she can use his/ her own private key to do so. Sender produces a hash value of the message, raises it to the power of d (modulo n) and attaches it as a "signature" to the message.

When receiver receives the signed message, he/she uses the same hash algorithm in conjunction with Alice's public key. Receiver raises the signature to the power of e (modulo n) and compares the resulting hash value with the message's actual hash value. If both of them are same, then receiver knows that the author of the message was in possession of sender's private key, and that the message has not been tampered since.

### 4.2 Hash Based Message Authentication Code (HMAC)

1) *Description:* The definition of HMAC requires a cryptographic hash function, were we signify by H, and a mystery key K. We accept H to be a cryptographic hash function where data is hashed by iterating a basic compression function on squares of information. We mean by B the byte-length of such blocks and by L the byte-length of hash outputs .The authentication key K can be of length up to B, the square length of the hash capacity.

Applications that utilize keys longer than B bytes will first hash the key using H and then use the resultant L byte string as the genuine key to HMAC. Regardless the

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

negligible prescribed length for K is L bytes. We define two fixed and different strings ipad and opad as follows (the 'i' and 'o' are mnemonics for inner and outer): ipad = the byte 0x36 repeated B times and opad = the byte 0x5C repeated B times. We can compute HMAC over the data `text' we perform H(K XOR opad, H(K XOR ipad, text)), append zeros to the end of K to create a B byte string, XOR (bitwise exclusive-OR) the B byte string with ipad, append the stream of data 'text' to the B byte string resulting then apply H to the stream generated before. XOR (bitwise exclusive-OR) the B byte string computed previously with opad then append the H result to the B byte string. Finally, apply H to the stream generated.

2) *Keys*: The key for HMAC can be of any length. However, less than L bytes is strongly discouraged as it would decrease the security strength of the function. So, Keys longer than L bytes are satisfactory but the extra length would not significantly increase the function quality. A more drawn out key may be prudent if the haphazardness of the key is considered weak. Keys need to be chosen at random and periodically refreshed. Periodic key refreshment is a fundamental security practice that helps against potential weaknesses of the function and keys, and limits the harm of an uncovered key.

3) *Implementation*: HMAC is defined in such a way that the underlying hash function H can be used with no adjustment to its code. Specifically, it utilizes the capacity H with the pre-defined initial value IV (a fixed value specified by each iterative hash function to initialize its compression function). The idea is that the intermediate results of the compression function on the B-byte blocks (K XOR ipad) and (K XOR opad) can be pre computed only once at the time of generation of the key K, or before its first utilize. These middle results are stored and then used to initialize the IV of H each time that a message needs to be confirmed.

This strategy spares, for every validated message the application of the compression function of H on two B-byte blocks. We stress that the stored intermediate values need to be treated and protected the same as secret keys. We recommend that the output length t be not less than half the length of the hash output and not less than 80 bits.

4) *Security:* A correct implementation of the above construction, the choice of random (or cryptographically pseudorandom) keys, a secure key exchange mechanism, frequent key refreshments, and good secret protection of keys are all essential ingredients for the security of the integrity verification mechanism provided by HMAC.

**4.3 Revocation**

We observed that in existing system that, even if a user is revoked from groups, he/she may still be able to access the data with the other attributes that he/she holds as long as they satisfy the access policy because they would still be effective in the system as they used CP-ABE Algorithm. So, in this paper we used RSA Encryption Algorithm with HMAC Algorithm to update the secret key for each group in particular time, and also update the new group key for each and every node in the groups to avoid intrusion. Now if node signature is not matched to the storage head signature while data transfer, then key update is again performed for secrecy and privacy.

**4.4 Proofs of Correctness:**

**4.4.1 Proof using Fermat's little theorem:** The proof of the correctness of RSA is based on Fermat's little theorem. This hypothesis expresses that if p is prime and p does not divide an integer a then $a^{p-1} \equiv 1 \pmod{p}$ they had shown that $m^{ed} \equiv m \pmod{pq}$ for every integer m when p and q are distinct prime numbers and e and d are positive integers satisfying $ed \equiv 1 \pmod{(p-1)(q-1)}$. It can be written as $ed - 1 = h(p-1)(q-1)$ for some non negative integer h. To check two numbers, like $m^{ed}$ and $m$, are congruent mod pq it is needed to check they are congruent mod p and mod q separately. To show $m^{ed} \equiv m \pmod{p}$, we consider two cases: $m \equiv 0 \pmod{p}$ and m not congruent 0 (mod p).

In the first case $m^{ed}$ is a various of p, so $m^{ed} \equiv 0 \equiv m \pmod{p}$. In the second case $m^{ed} = m^{(ed-1)}m = m^{h(p-1)(q-1)}m = (m^{p-1})^{h(q-1)}m \equiv 1^{h(q-1)}m \equiv m \pmod{p}$ where we used Fermat's little theorem to replace $m^{p-1} \bmod p$ with 1. The verification that $m^{ed} \equiv m \pmod{q}$ proceeds in a similar way, treating separately the cases $m \equiv 0 \pmod{q}$ and $m$ not congruent to 0 (mod $q$), utilizing Fermat's little theorem for modulus $q$ in the second case This completes the confirmation that, for any m, $(m^e)^d \equiv m \pmod{pq}$.

**4.4.2 Proof using Euler's theorem:** We want to show that $m^{ed} \equiv m \pmod{n}$, where $n = pq$ is a product of two different prime numbers and $e$ and $d$ are positive integers satisfying $ed \equiv 1 \pmod{\varphi(n)}$. Since $e$ and $d$ are positive, we can compose $ed = 1 + h\varphi(n)$ for some non-negative whole number h. Expecting that m is moderately prime to n, we have $m^{ed} = m^{1+h\varphi(n)} = m(m^{\varphi(n)})^h \equiv m(1)^h \equiv m \pmod{n}$ where the second-last congruence follows from the Euler's theorem. When $m$ is not moderately prime to n, the contention simply given is invalid. This is highly anyhow even for this situation the craved harmoniousness is still true. Either $m \equiv 0 \pmod{p}$ or $m \equiv 0 \pmod{q}$, and these cases can be treated using the previous proof.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

## 5. CONCLUSION

Rivest Shamir Adleman RSA) Algorithm is an encryption algorithm where every member of a group will be provided separate keys for encryption and decryption where different users are allowed to decrypt different pieces of message as per their security policy. In this paper, we proposed an efficient and secure data retrieval method using RSA Algorithm for mobilised DTN's where central key authority manages their attributes strictly and secretly. The issue of same key to users is resolved here such that the confidentiality is maintained at extreme level. In addition, the HMAC Algorithm is used for the signature based revocation for protection. We demonstrate how to apply the proposed mechanism such that only authorized mobile nodes can securely and effectively send and retrieve the confidential data in the disruption-tolerant military network.

## References

**[1]** **J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: 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. Ammar, and E. Zequra, "Mesage 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 ciphertext 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]** **J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute based encryption," in *Proc. IEEE Sims. Security Privacy*, 2007, pp. 321–334.**

**[12]** **A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in *Proc. ACM Conf. Comput. Commun. Security*, 2008, pp. 417–426.**

**[13]** **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.**