ISSN:0975-9646

Venkateswara Dunna et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (6) , 2015, 5238-5243

A Novel Secured Data Transmission using Less Number of Keys in Sensor Nodes Used in Security Application Networks

Venkateswara Dunna[1] Y.Ramesh kumar[2]
[1]*M.Tech CSE,* [2]*Associate Professor,*
[1,2]*Department of CSE, Avanthi Institute of Engineering and Technology (JNTUK)*

Abstract: Sensor 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. In this paper, we propose a secure data retrieval scheme using CP-ABE for decentralized DTNs where minimum number of multiple key authorities manage their attributes independently by using mutual authentication protocol, data exchange protocol and symmetric key block encryption algorithm. We demonstrate how to apply the proposed mechanism to securely and efficiently manage the confidential data distributed in the disruption-tolerant military network

Keywords: encryption,decryption,encoding,decoding.

INTRODUCTION

In this paper we tried to implement security in a Network using cipher text-ABE and Key policy-ABE. This project is embedded with various Server applications and used to encrypt and decrypt the information that is being sent and informing about the server when an intrusion is detected.

Decentralized CP-ABE schemes in the multi-authority network environment. They achieved a combined access policy over the attributes issued from different authorities by simply encrypting data multiple times. The main disadvantages of this approach are efficiency and expressiveness of access policy. For example, when a commander encrypts a secret mission under the policy ("Battalion 1" AND ("Region 2" OR 'Region 3")), it cannot be expressed when each "Region" attribute is managed by different authorities, since simply multi encrypting approaches

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, even if each of them cannot decrypt it individually. We do not want these colluders to be able to decrypt the secret information by combining their attributes. We also consider collusion attack among curious local authorities to derive users' keys.

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.

- The concept of attribute-based encryption (ABE) is a promising approach that fulfills the requirements for secure data retrieval in DTNs.
- ABE features a mechanism that enables an access control over encrypted data using access policies and a scribed attributes among private keys and ciphertexts.
- 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 de- crypt the ciphertext .
- 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.

ABE comes in two flavors called key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE). In KP-ABE, the encryptor only gets to label a ciphertext with a set of attributes.

Attribute Revocation:

A user who newly holds the attribute might be able to access the previous data encrypted before obtains the attribute until the data is re-encrypted with the newly updated attribute keys by periodic rekeying.

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

Decentralized ABE:
Achieved a combined access policy over the attributes issued from different authorities by simply encrypting data multiple times. The main disadvantages of this approach are efficiency and expressiveness of access policy.

Encryption and Decryption of data:
In cryptography, **Encryption** is the process of encoding messages or information in such a way that only authorized parties can read it. In an encryption scheme, the message or information, referred to as plain-text, is encrypted using an encryption algorithm [11], turning it into an unreadable cipher text [6].

Decryption: It is the process of decoding the data which has been encrypted into a secret format. An authorized user can only decrypt data because decryption requires a secret key or password. In simple terms it is the conversion of cipher text into plain text [6].

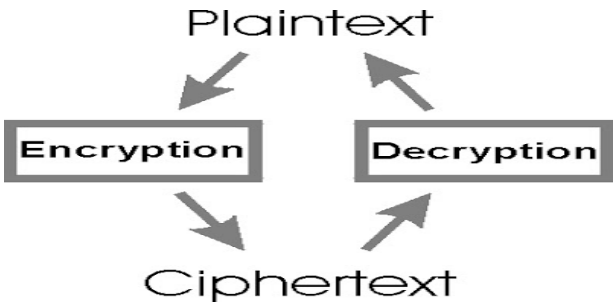


Figure 1: Conversion of plain text to cipher text and vice versa

Compression and Encoding algorithms are of major interest of the research since long. The following algorithms are a few of them are briefed below:

Triple DES
Triple DES (3DES) is the common name for the Triple Data Encryption Algorithm (TDEA) block cipher, which applies the Data Encryption Standard (DES) cipher algorithm three times to each data block. Because of the availability of increasing computational power, the key size of the original DES cipher was becoming subject to brute force attacks; Triple DES was designed to provide a relatively simple method of increasing the key size of DES to protect against such attacks, without designing a completely new block cipher algorithm. ***This suffers from time complexity and long ciphers to be transmitted.***

AES
The Advanced Encryption Standard (AES) is a symmetric-key encryption standard adopted by the U.S. government. The standard comprises three block ciphers, AES-128, AES-192 and AES-256, adopted from a larger collection originally published as Rijndael. Each of these ciphers has a 128-bit block size, with key sizes of 128, 192 and 256 bits, respectively. ***This also suffers from time complexity and long ciphers to be transmitted.***

RC4
RC4 (also known as ARC4 or ARCFOUR meaning Alleged RC4, see below) is the most widely-used software stream cipher and is used in popular protocols such as Secure Sockets Layer (SSL) (to protect Internet traffic) and WEP (to secure wireless networks). ***This is less computationally intensive compared to asynchronous algorithms.*** *Thus, an attempt to construct a simple, faster, and less heavy cipher encoder algorithm is being made in this project.*

METHODOLOGY

Let n denote the number of sensors in our network. Without loss of generality, we assume that n is an odd positive integer. Each sensor in the network has a unique identifier in the range 0: n - 1. We use ix and iy to denote the identifiers of sensors x and y, respectively, in this network.
Each two sensors, say sensors x and y, share a symmetric key denoted Kx;y or Ky;x. Only the two sensors x and y know their shared key Kx;y. And if sensors x and y ever become neighbours in the network, then they can use their shared symmetric key Kx;y to perform two functions:

1) Mutual Authentication: Sensor x authenticates sensor y, and sensor y authenticates sensor x.

2) Confidential Data Exchange: Encrypt and later decrypt all the exchanged data messages between x and y. (Note that sensors x and y can become neighbours in the network in two occasions. First, the two sensors x and y could be mobile and their movements cause them to become adjacent to one another. Second, the two sensors could be stationary and they are deployed adjacent to one another.)

In the remainder of this section, we show that if the shared symmetric keys are designed to have a "special structure", then each sensor needs to store only (n+1)/2 shared symmetric keys. But before we present the special structure of the shared keys, we need to introduce two new concepts: "universal keys" and "a circular relation, named below, over the sensor identifiers".

Each sensor x in the network stores a symmetric key, called the universal key of sensor x. The universal key of sensor x, denoted ux, is known only to sensor x.

Let ix and iy be two distinct sensor identifiers. (Recall that both ix and iy are in the range 0: n-1, where n is the odd number of sensors in the sensor network.) Identifier ix is said to be below identifier iy if exactly one of the following two conditions holds:

1) ix<iy and (iy - ix) < n=2
2) ix>iy and (ix - iy) > n=2

The below relation is better explained by an example. Consider the case where n = 5. In this case, the sensor identifiers are 0, 1, 2, 3, and 4, and we have:
Identifier 0 is below identifiers 1 and 2.
Identifier 1 is below identifiers 2 and 3.
Identifier 2 is below identifiers 3 and 4.
Identifier 3 is below identifiers 4 and 0.
Identifier 4 is below identifiers 0 and 1

A MUTUAL AUTHENTICATION PROTOCOL:

Before the sensors are deployed in a network, each sensor x is supplied with the following items:
1) One distinct identifier ix in the range 0: n-1
2) One universal key ux
3) (n-1)/2 symmetric keys Kx;y = H(ix,uy) each of which is shared between sensor x and another sensor y, where ix is below iy After every sensor is supplied with these items,

the sensors are deployed in random locations in the network.

Now if two sensors x and y happen to become adjacent to one another, then these two sensors need to execute a mutual authentication protocol so that sensor x proves to sensor y that it is indeed sensor x and sensor y proves to sensor x that it is indeed sensor y.

The mutual authentication protocol consists of the following six steps.

Step 1: Sensor x selects a random nonce nx and sends a hello message that is received by sensor y.

x → y : hello(ix, nx)

Step 2: Sensor y selects a random nonce ny and sends a hello message that is received by sensor x.

x ← y : hello(iy, ny)

Step 3: Sensor x determines whether ix is below iy. Then it either fetches Kx;y from its memory or computes it. Finally, sensor x sends a verify message to sensor y.

x → y : verify(ix, iy, H(ix iynyKx;y))

Step 4: Sensor y determines whether iy is below ix. Then it either fetches Kx;y from its memory or computes it. Finally, sensor y sends a verify message to sensor x.

x ← y : verify(iy, ix, H(iy ix nxKx;y))

Step 5: Sensor x computes H(iy ix nx*Kx;y) and compares it with the received H(iy ix nx*Kx;y). If they are equal, then x concludes that the sensor claiming to be sensor y is indeed sensor y. Otherwise, no conclusion can be reached.

Step 6: Sensor y computes H(ix, iy,ny,Kx;y) and compares it with the received H(ix, iy,ny,Kx;y). If they are equal, then y concludes that the sensor claiming to be sensor x is indeed sensor x. Otherwise, no conclusion can be reached.

A DATA EXCHANGE PROTOCOL:

After two adjacent sensors x and y have authenticated one another using the mutual authentication protocol described in the previous section, sensors x and y can now start exchanging data messages according to the following data exchange protocol. (Recall that nx and ny are the two nonce's that were selected at random by sensors x and y, respectively, in the mutual authentication protocol.)

Step 1: Sensor x concatenates the nonce ny with the text of the data message to be sent, encrypts the concatenation using the symmetric key Kx;y, and sends the result in a data message to sensor y.

x → y : data(ix, iy, Kx,y (ny text))

Step 2: Sensor y concatenates the nonce nx with the text of the data message to be sent, encrypts the concatenation using the symmetric key Kx;y, and sends the result in a data message to sensor x.

x ← y : data(iy, ix, Kx,y (nx text))

Sensors x and y can repeat Steps 1 and 2 any number of times to exchange data between themselves.

Symmetric key block encryption algorithm:

Algorithm:

High-level description of the algorithm

- KeyExpansion—round keys are derived from the cipher key using Rijndael's key schedule
- Initial Round

1. AddRoundKey—each byte of the state is combined with the round key using bitwise xor

- Rounds

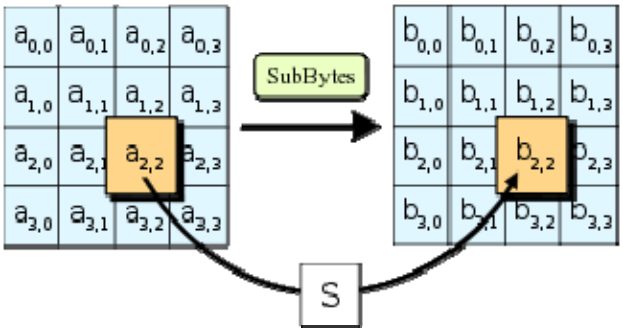
1. SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.
2. ShiftRows—a transposition step where each row of the state is shifted cyclically a certain number of steps.
3. MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column
4. AddRoundKey

- Final Round (no MixColumns)

1. SubBytes
2. ShiftRows
3. AddRoundKey

The SubBytes step

In the SubBytes step, each byte in the state is replaced with its ntry in a fixed 8-bit lookup table, *S*; $b_{ij} = S(a_{ij})$.In the SubBytes step, each byte in the array is updated using an 8-bit substitution box, the Rijndael S-box. This operation provides the non-linearity in the cipher. The S-box used is derived from the multiplicative inverse over **GF**(2^8), known to have good non-linearity properties. To avoid attacks based on simple algebraic properties, the S-box is constructed by combining the inverse function with an invertible affine transformation. The S-box is also chosen to avoid any fixed points (and so is a derangement), and also any opposite fixed points.



The Shift Rows step

In the ShiftRows step, bytes in each row of the state are shifted cyclically to the left. The number of places each byte is shifted differs for each row.

The Shift Rows step operates on the rows of the state; it cyclically shifts the bytes in each row by a certain offset. For AES, the first row is left unchanged. Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively. For the block of size 128 bits and 192 bits the shifting pattern is the same. In this way, each column of the output state of the Shift Rows step is composed of bytes from each column of the input state. (Rijndael variants with a larger block size have slightly different offsets). In the case of the 256-bit block, the first row is unchanged and the shifting for second, third and fourth row is 1 byte, 3 bytes and 4 bytes respectively - this change only applies for the Rijndael cipher when used with a 256-bit block, as AES does not use 256-bit blocks.

In the Mix Columns step, the four bytes of each column of the state are combined using an invertible linear

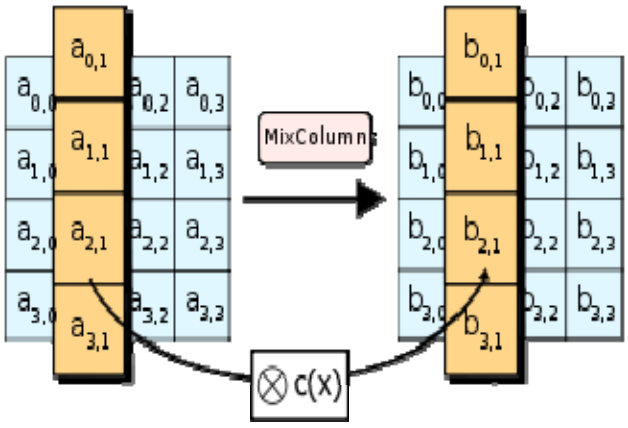
transformation. The Mix Columns function takes four bytes as input and outputs four bytes, where each input byte affects all four output bytes. Together with Shift Rows, Mix Columns provides diffusion in the cipher.

During this operation, each column is multiplied by the known matrix that for the 128 bit key is

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}.$$

The multiplication operation is defined as: multiplication by 1 means leaving unchanged, multiplication by 2 means shifting byte to the left and multiplication by 3 means shifting to the left and then performing xor with the initial un-shifted value.

In more general sense, each column is treated as a polynomial over GF(2^8) and is then multiplied modulo x^4+1 with a fixed polynomial $c(x) = 0x03 \cdot x^3 + x^2 + x + 0x02$. The coefficients are displayed in their hexadecimal equivalent of the binary representation of bit polynomials from GF(2)[x]. The Mix Columns step can also be viewed as a multiplication by a particular MDS matrix in a finite field. This process is described further in the article Rijndael mix columns.



The Add Round Key step

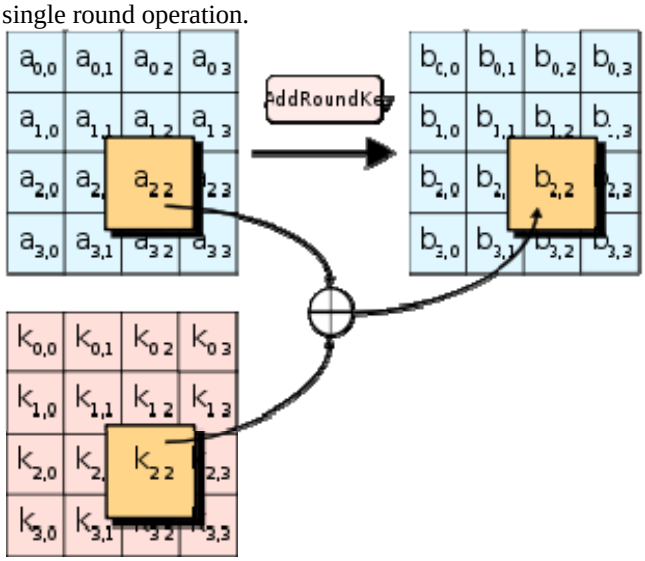
In the Add Round Key step, the sub key is combined with the state. For each round, a sub key is derived from the main key using Rijndael's key schedule each sub key is the same size as the state. The sub key is added by combining each byte of the state with the corresponding byte of the sub key using bitwise XOR.

Optimization of the cipher

On systems with 32-bit or larger words, it is possible to speed up execution of this cipher by combining SubBytes and Shift Rows with Mix Columns, and transforming them into a sequence of table lookups. This requires four 256-entry 32-bit tables, which utilizes a total of four kilobytes (4096 bytes) of memory—one kilobyte for each table. A round can now be done with 16 table lookups and 12 32-bit exclusive-or operations, followed by four 32-bit exclusive-or operations in the Add Round Key step

If the resulting four kilobyte table size is too large for a given target platform, the table lookup operation can be performed with a single 256-entry 32-bit (i.e. 1 kilobyte) table by the use of circular rotates.

Using a byte-oriented approach, it is possible to combine the Sub Bytes, Shift Rows, and Mix Columns steps into a single round operation.



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.

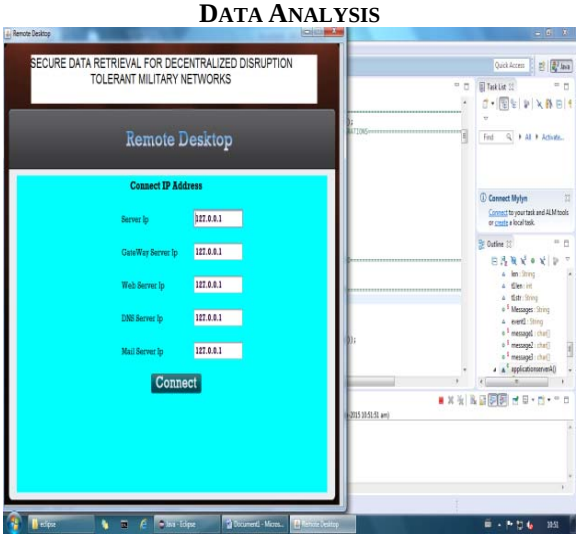
Sender:

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.

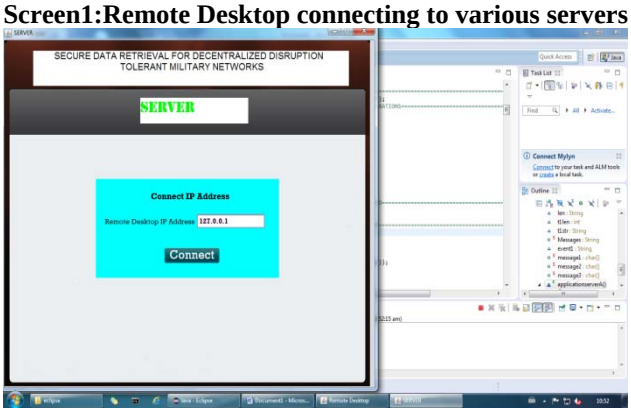
User:

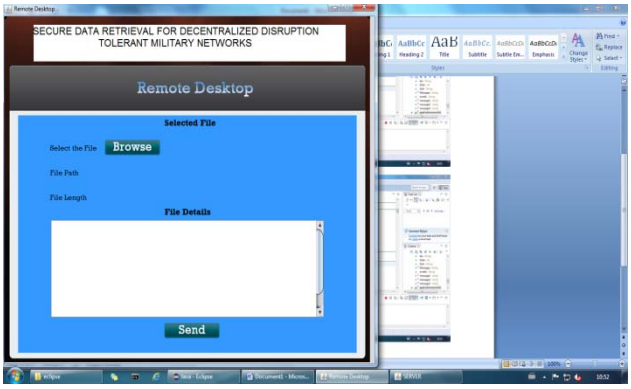
This is a mobile node who wants to access the data stored at the storage node (e.g., a soldier). If a user possesses a set of attributes satisfying the access policy of the encrypted data defined by the sender, and is not revoked in any of the attributes, then he will be able to decrypt the ciphertext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

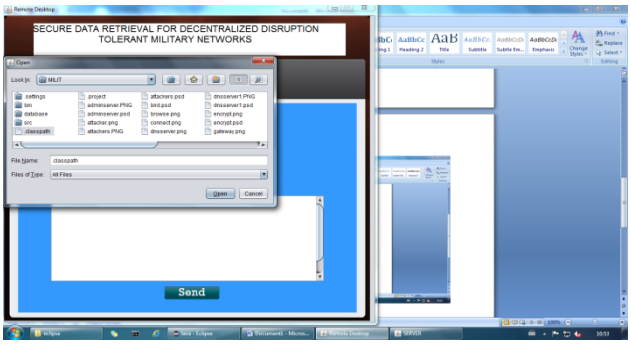
DATA ANALYSIS

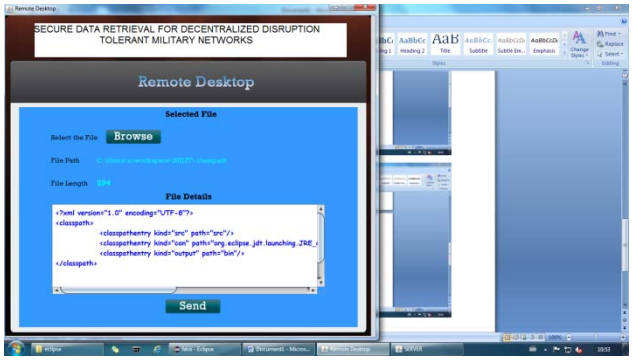


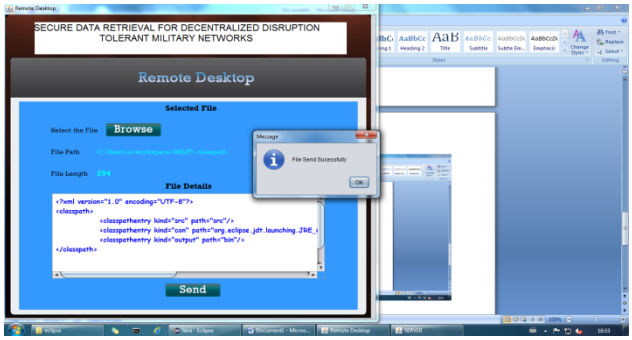
Screen1:Remote Desktop connecting to various servers

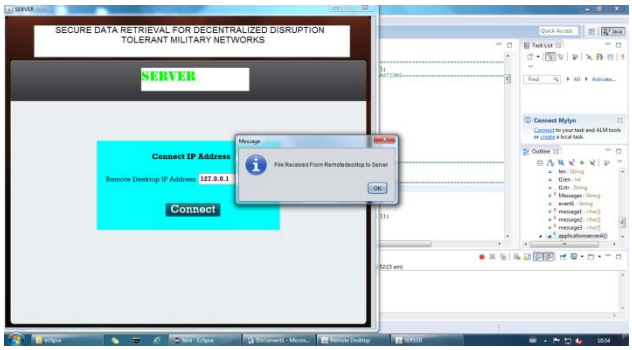


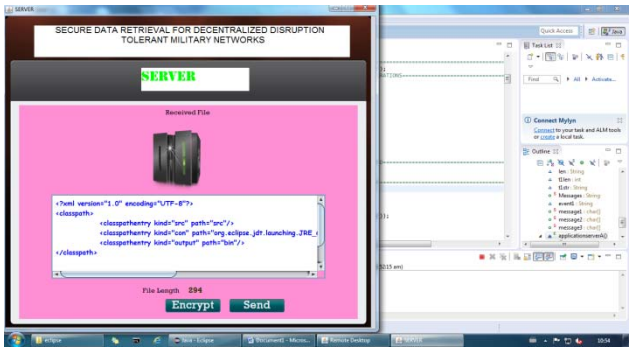


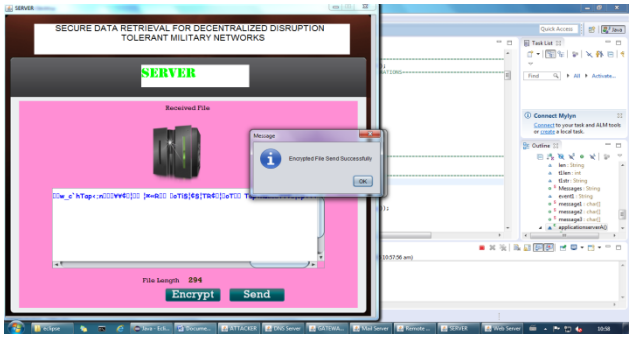


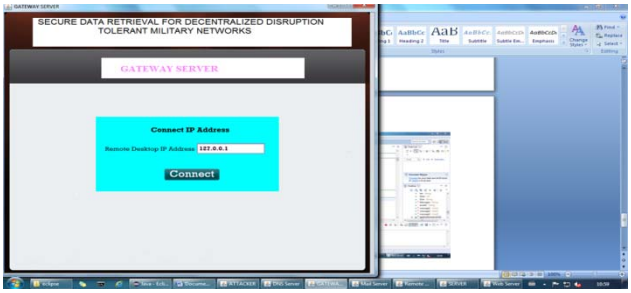


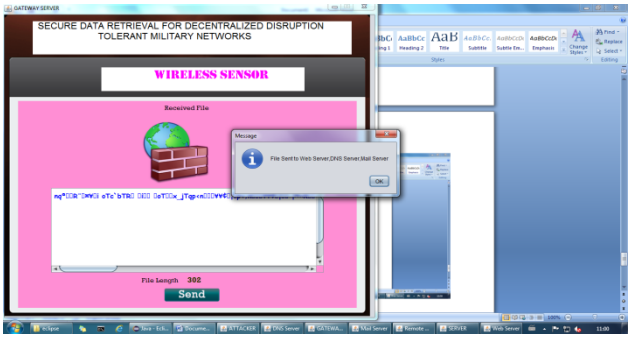












CONCLUSION

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 dis- ruption-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. ACMMobiHoc*, 2006, pp. 37–48.
4. S. Roy andM. 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 *ProcConf. 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 attribute based 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 in multi-authority 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.
27. M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya, "P-signatures and noninteractive anonymous credentials," in *Proc. TCC*, 2008, LNCS 4948, pp. 356–374.
28. M.Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss,A.Hysyanskaya, and H. Shacham, "Randomizable proofs and delegatable anonymous credentials," in *Proc. Crypto*, LNCS 5677, pp. 108–125.
29. D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Proc. CRYPTO*, 2001, LNCS 2139, pp. 41–62.
30. C. K.Wong,M. Gouda, and S. S. Lam, "Secure group communications using key graphs," in *Proc. ACM SIGCOMM*, 1998, pp. 68–79.
31. A. T. Sherman and D. A. McGrew, "Key establishment in large dynamic groups using one-way function trees," *IEEE Trans. Softw. Eng.*, vol. 29, no. 5, pp. 444–458, May 2003.
32. K. C. Almeroth and M. H. Ammar, "Multicast group behavior in the Internet's multicast backbone (MBone)," *IEEE Commun. Mag.*, vol. 35, no. 6, pp. 124–129, Jun. 1997.
33. "The Pairing-Based Cryptography Library," Accessed Aug. 2010 [Online]. Available: http://crypto.stanford.edu/pbc/