Accessing Secured Data using Two Level Encryption in Decentralized Disruption-Tolerant Military Networks

Nethravathi B M
MTech (CSE)
TJIT, Bangalore

Anita B
Asst.Professor, Dept of CSE
TJIT, Bangalore

V Sherin George
Asst.Professor, Dept of CSE
TJIT,Bangalore

***Abstract*— Nodes in a battlefield creates problems like network connectivity failure and frequent partitioning as the nodes are mobile in nature. Disruption-tolerant network (DTN) technologies are playing a crucial role in overcoming the above problems. It allows wireless devices to be carried out by soldiers to communicate among themselves and also with the commander so that the confidential information is forwarded reliably by using external storage nodes in the network. Sending secured and confidential files to the authorized people are the most challenging issue in this scenario. However, Ciphertext-spolicy attribute-based encryption (CP-ABE) in decentralized DTNs introduces many security and privacy issues with respect to the attribute updation, key escrow, and therefore to overcome the issues, we proposed an attribute-based secure data retrieval scheme using CP-ABE for decentralized DTNs.**

In the Proposed mechanism, the key issuing protocol generates keys using the two-party computation (2PC) protocol from the key authorities with their own master secrets and issues user secret. The 2PC protocol prevents the key authorities from obtaining any master secret information of each other such that both the parties generate the half part of the keys. And to get the original message, it depends on two keys, one from key authorities and another from the storage node. Thus, users are not required to fully trust the authorities in order to protect their data to be shared. The data confidentiality and privacy can be cryptographically enforced against any curious key authorities or data storage nodes in the proposed scheme.

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

I. INTRODUCTION

In many military networks, connections of wireless devices carried by soldiers may be temporarily disconnected by jamming, environmental factors, and mobility, especially when they operate in hostile environments. Disruption-tolerant network (DTN) technologies are day by day becoming one of the successful solutions to overcome these existing problems and allow nodes to communicate with each other in these extreme networking environments [1]–[3].When there is no end-to-end connection between a source and a destination pair in the network, the messages from the source node may need to wait in the intermediate nodes until it finds a path towards the destination. Roy [4] and Chuah [5] 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 [6], [7]. In many cases, it is desirable to provide differentiated access services such that data access policies are defined over user attributes or roles, which are managed by the key authorities. For example, in a disruption-tolerant military network, a commander may store confidential information at a storage node, which should be accessed by members of "Battalion 1" who are participating in "Region 2." In this case, it is a reasonable assumption that multiple key authorities are likely to manage their own dynamic attributes for soldiers in their deployed regions or echelons, which could be frequently changed (e.g., the attribute representing current location of moving soldiers) [4], [8], [9]. We refer to this DTN architecture where multiple authorities issue and manage their own attribute keys independently as a decentralized DTN [10].

The concept of attribute-based encryption (ABE) [11]–[14] 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 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 [13]. 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. If the key authority is compromised by adversaries when deployed in the hostile environments, this could be a potential threat to the data confidentiality or privacy especially when the data is highly sensitive. The key escrow is an inherent problem even in the multiple-authority systems as long as each key authority has the whole privilege to generate their own attribute keys with their own master secrets. Since such a key generation mechanism based on the single master secret is the basic method for most of the asymmetric encryption systems such as the attribute- based or identity-based encryption protocols, removing escrow in single or multiple-authority CP-ABE is a pivotal open problem.

The last challenge is the coordination of attributes issued from different authorities. When multiple authorities manage and issues 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.

II. RELATED WORK

In paper[1] author discusses on the Group communication.It benefits from IP multicast to achieve scalable exchange of messages,but it did not prevent the non goup users.So encryption was used to protect messages.

In paper[2] author concentrates on stateless receiver case where the users do not update their state from session to session. So they present a framework called the subset cover frame,which abstracts a variety of revocation schemes.

In paper[3] author assumes regular nodes volunteer to be message ferries when network dynamics mandate the presence of such ferries to ensure communications. Thus, a node-density based adaptive routing (NDBAR) scheme allows regular nodes to volunteer to be message ferries when there are very few nodes around them to ensure the feasibility of continued communications.

In paper[4] author designs MaxProp, a protocol for effective routing of DTN messages. MaxProp is based on prioritizing both the schedule of packets transmitted to other peers and packets to be dropped. These priorities are based on the historical data. Max Prop performs better than protocols that have access to an oracle that knows the schedule of meetings between peers.

In paper[5] author says about CP-ABE that allows a new type of encrypted access control.In this user's private keys are specified by a set of attributes and a party encrypting data can specify a policy over these attributes specifying which users are able to decrypt.

In paper[6] P-signature scheme consists of a signature scheme that incudes an interactive protocol for obtaining a signature on a committed value, A non-interactive proof system for proving that the contents of a commitment has been signed.This serve as a useful building block for other privacy-preserving.

In paper[7] Introduced a secure information management architecture based on attribute-based encryption (ABE) primitives. Proposed cryptographic optimizations that improves the efficiency in a distributed environment.

In paper[8] Plutus is a cryptographic storage system that make use of cryptographic primitives to protect and share files. Plutus features highly scalable key management. Stored data is encrypted and key management will be done by the client.

A.ABE

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.The key authority chooses a policy for each user that determines which ciphertext he can decrypt and issues the key to each user by embedding the policy into the user's key. However, the roles of the ciphertexts and keys are reversed in CP-ABE. In CP-ABE, the ciphertext is encrypted with an access policy chosen by an encryptor, 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 encryptors 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 [4], [7], and [15].

1) *Attribute Revocation*: Bethencourt et al. [13] and Boldyreva et al. [16] 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 [8], [13], [16], [7] have two main problems. The first problem is the security degradation in terms of the backward and forward secrecy [15]. It is a considerable scenario that users such as soldiers may change their attributes frequently, e.g., position or location move when considering these as attributes [4], [9]. Then, a user who newly holds the attribute might be able to access the previous data encrypted before he that users such as soldiers may change their attributes frequently, e.g., position or location move when considering these as attributes [4], [9]. 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 reencrypted with the newly updated attribute keys by periodic rekeying (backward secrecy). For example, assume that at time, a ciphertext is encrypted with a policy that can be decrypted with a set of

attributes (embedded in the user's keys) for users with. After time, say, a user newly holds the attribute set. Even if the new user should be disallowed to decrypt the ciphertext for the time instance, he can still decrypt the previous ciphertext until it is reencrypted with the newly updated attribute keys. On the other hand, a revoked user would still be able to access the encrypted data even if he does not hold the attribute any more until the next expiration time (forward secrecy). For example, when a user is disqualified with the attribute at time, he can still decrypt the ciphertext of the previous time instance unless the key of the user is expired and the ciphertext is re 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 non revoked users can update their keys. This results in the "1-affects-" problem, which means that the update of a single attribute affects the whole non revoked users who share the attribute [13]. This could be a bottleneck for both the key authority and all non revoked users. The immediate key revocation can be done by revoking users using ABE that supports negative clauses [4], [14]. To do so, one just adds conjunctively the AND of negation of revoked user identities (where each is considered as an attribute here). However, this solution still somewhat lacks efficiency performance. This scheme will pose overhead group elements1 additively to the size of the ciphertext and multiplicatively to the size of private key over the original CP-ABE scheme of Bethencourt et al. [13], where is the maximum size of revoked attributes set. Golle et al. [10] also proposed a user revocable KP-ABE scheme, but their scheme only works when the number of attributes associated with a ciphertext is exactly half of the universe size.

2) *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 [11], [13], [14]. Thus, the key escrow problem is inherent such that the key authority can decrypt every ciphertext addressed to users in the system by generating their secret keys at any time. Chase et al. [14] 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 result in communication overhead on the system setup and the rekeying phase's 1The group elements mean those in the pairing operation group, not the user group. Since the computation in ABE schemes is done in the pairing operation group, the group element in the manuscript means group elements in the pairing group.

3) *Decentralized ABE*: Huang et al. [9] and Roy et al. [4] proposed 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 to soldiers 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 can by no means express any general " -out-of- " logics (e.g., OR, that is 1-out-of-). For example, let $A_1...A_N$ be the key authorities, and $a_1 \ldots a_N$ be attributes sets they independently manage, respectively. Then, the only access policy expressed with is, which can be achieved by encrypting a message with $a_1 \ldots a_N$ is (a_1 AND…AND a_N), and then encrypting the resulting ciphertext with a_1 by A_1, and then encrypting resulting ciphertext with c_1 with a_2 by A_2, and so on, until this multiencryption generates the final ciphertext . 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 [15] and Lewko et al. [10] proposed multiauthority KP-ABE and CP-ABE schemes, respectively. However, their schemes also suffer from the key escrow problem like the prior decentralized schemes.

B. Diffie–Hellman key exchange

Diffie–Hellman key exchange (D–H) is a specific method of securely exchanging cryptographic keys over a public channel and was the first specific example of public-key cryptography as originally conceptualized. D–H is one of the earliest practical examples of public key exchange implemented within the field of cryptography. The Diffie–Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communication channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher. Diffie–Hellman key agreement itself is an anonymous (non-authenticated) key-agreement protocol, it provides the basis for a variety of authenticated protocols, and is used to provide perfect forward secrecy in Transport Layer Security's ephemeral modes.

Diffie–Hellman establishes a shared secret that can be used for secret communications while exchanging data over a public network. The following diagram illustrates the general idea of the key exchange by using colors instead of a very large number. The crucial part of the process is that Alice and Bob exchange their secret colors in a mix only. Finally this generates an identical key that is computationally difficult (impossible for modern supercomputers to do in a reasonable amount of time) to reverse for another party that might have been listening in on them. Alice and Bob now use this common secret to encrypt and decrypt their sent and received data. The starting color (yellow) is arbitrary, but is agreed on in advance by Alice and Bob, and does not need to be secret.

C. Cryptographic explanation

The simplest and the original implementation of the protocol uses the multiplicative group of integers modulo p, where p is prime, and g is a primitive root modulo p. Here is an example of the protocol, with non-secret values in blue and secret values in red.

1. Alice and Bob agree to use a prime number $p = 23$ and base $g = 5$ (which is a primitive root modulo 23).
2. Alice chooses a secret integer $a = 6$, then sends Bob $A = g^a \bmod p$
   - $A = 5^6 \bmod 23 = 8$
3. Bob chooses a secret integer $b = 15$, then sends Alice $B = g^b \bmod p$
   - $B = 5^{15} \bmod 23 = 19$
4. Alice computes $s = B^a \bmod p$
   - $s = 196 \bmod 23 = 2$
5. Bob computes $s = A^b \bmod p$.
   - $s = 815 \bmod 23 = 2$.
6. Alice and Bob now share a secret (the number 2).

Both Alice and Bob have arrived at the same value, because $(g^a)^b$ (for Bob, 815 mod 23 = (g^a mod p)b mod p = $(g^a)^b$ mod p) and $(g^b)^a$ are equal mod p. Note that only a, b, and (g^{ab} mod p = g^{ba} mod p) are kept secret. All the other values – p, g, g^a mod p, and g^b mod p – are sent in the clear. Once Alice and Bob compute the shared secret they can use it as an encryption key, known only to them, for sending messages across the same open communications channel. Of course, much larger values of a, b, and p would be needed to make this example secure, since there are only 23 possible results of n mod 23. However, if p is a prime of at least 300 digits, and a and b are at least 100 digits long, then even the fastest modern computers cannot find a given only g, p, g^b mod p and g^a mod p. The problem such a computer needs to solve is called the discrete logarithm problem. The computation of g^a mod p is known as modular exponentiation and can be done efficiently even for large numbers. Note that g need not be large at all, and in practice is usually a small prime (like 2, 3, 5...) because primitive roots usually are quite numerous.

D. Generalization to finite cyclic groups:

Here's a more general description of the protocol,

1. Alice and Bob agree on a finite cyclic group G and a generating element g in G. (This is usually done long before the rest of the protocol; g is assumed to be known by all attackers.) We will write the group G multiplicatively.
2. Alice picks a random natural number a and sends g^a to Bob. 3. Bob picks a random natural number b and sends g^b to Alice.
4. Alice computes $(g^b)^a$.
5. Bob computes $(g^a)^b$.

Both Alice and Bob are now in possession of the group element g^{ab}, which can serve as the shared secret key.

If m is a message, and an element of the group, then we can encrypt $e = mg^{ab}$. Then we can decipher m from e as follows: We compute $(g^{ab})^{-1}$, using |G|:

Bob knows G, b, and g^a. As g generates G it follows that $g^{|G|} = 1$ (the group identity).

Bob calculates $(g^a)^{|G|-b} = g^{a(|G|-ab} = g^{a|G|}g^{-ab}$–ab $=(g^{|G|})^a g^{-ab} = 1^a g^{-ab}=(g^{ab})^{-1}$.

When Alice sends Bob the encrypted message, $e = mg^{ab}$, Bob computes $e(g^{ba})^{-1} = mg^{ab}(g^{ab})^{-1} = m(1) = m$.

F. Secrecy chart

The chart below depicts who knows what, again with non-secret values in blue and secret values in red. Here Eve is an eavesdropper—she watches what is sent between Alice and Bob, but she does not alter the contents of their communications.

- g = public (prime) base, known to Alice, Bob, and Eve. $g = 5$
- p = public (prime) number, known to Alice, Bob, and Eve. $p = 23.S$
- a = Alice's private key, known only to Alice. $a = 6$.
- b = Bob's private key known only to Bob. $b = 15$.
- A = Alice's public key, known to Alice, Bob, and Eve. $A = g^a \bmod p = 8$
- B = Bob's public key, known to Alice, Bob, and Eve. $B = g^b \bmod p = 19$

E. Operation with more than two parties

Diffie–Hellman key agreement is not limited to negotiating a key shared by only two participants. Any number of users can take part in an agreement by performing iterations of the agreement protocol and exchanging intermediate data (which does not itself need to be kept secret). For example, Alice, Bob, and Carol could participate in a Diffie–Hellman agreement as follows, with all operations taken to be modulo:

The parties agree on the algorithm parameters and

1. The parties generate their private keys, named, and
2. Alice computes and sends it to Bob.
3. Bob computes and sends it to Carol.
4. Carol computes and uses it as her secret.
5. Bob computes and sends it to Carol.
6. Carol computes and sends it to Alice.
7. Alice computes and uses it as her secret.
8. Carol computes and sends it to Alice.
9. Alice computes and sends it to Bob.
10. Bob computes and uses it as his secret.

G. *Motivation*

In this paper, we propose a two level encryption and an attribute-based secure data retrieval scheme in DTN using CP-ABE. The proposed scheme features the following achievements. First, immediate attribute revocation enhances backward/forward secrecy of confidential data by reducing the windows of vulnerability. Second, encryptors can define a fine-grained access policy using any monotone access structure under attributes issued from any chosen set of authorities. Third, the key escrow problem is resolved by an escrow-free key issuing protocol that exploits the characteristic of the decentralized DTN architecture. The key issuing protocol generates and issues user secret keys by performing a secure two-party computation (2PC) protocol

among the key authorities with their own master secrets. The 2PC protocol deters the key authorities from obtaining any master secret information of each other such that none of them could generate the whole set of user keys alone. Thus, users are not required to fully trust the authorities in order to protect their data to be shared. The data confidentiality and privacy can be cryptographically enforced against any curious key authorities or data storage nodes in the proposed scheme.

III. NETWORK ARCHITECTURE

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

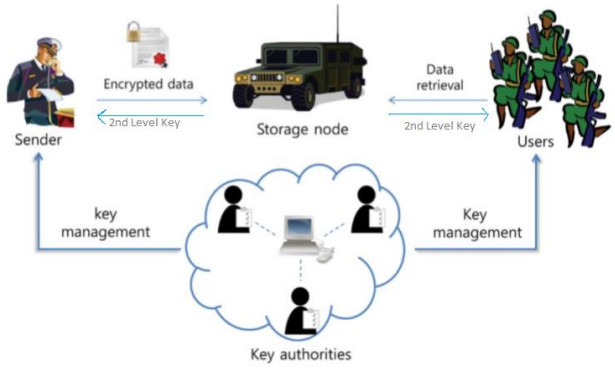

Fig1.System Architecture

A. System Description and Assumptions

Fig. 1 shows the architecture of the DTN. As shown in Fig. 1, the architecture consists of the following system entities.

1) 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.

2) Storage node: This is an entity that stores data from senders and provide corresponding access to users. It may be mobile or static [4], [5]. Similar to the previous schemes, we also assume the storage node to be semi trusted that is honest-but-curious. The second level key (SLK) is generated by the storage node. Commander before sending the first level encrypted message to the storage node sends a request to the storage node for the SLK and provide the details of the users who can access the message and in return the storage node sends the key to the commander and stores the key for the users while decrypting the message.

3) 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. Commander before sending the first level encrypted message to the storage node sends a request to the storage node for the SLK and encrypt the first level encrypted message using the SLK provided by the storage node.

4) User: This is a mobile node who wants to access the data stored at the storage node (e.g., a soldier). User sends the request to the storage node for the SLK and on receiving the SLK user decrypts the message. 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. Since the key authorities are semi-trusted, they should be deterred from accessing plaintext of the data in the storage node; meanwhile, they should be still able to issue secret keys to users. In order to realize this somewhat contradictory requirement, the central authority and the local authorities engage in the arithmetic 2PC protocol with master secret keys of their own and issue independent key components to users during the key issuing phase. The 2PC protocol prevents them from knowing each other's master secrets so that none of them can generate the whole set of secret keys of users individually. Thus, we take an assumption that the central authority does not collude with the local authorities (otherwise, they can guess the secret keys of every user by sharing their master secrets).

Algorithm 1 Token Pre-Computation

1: procedure
2: Choose parameters l, n and function f, ϕ;
3: Choose the number t of tokens;
4: Choose the number r of indices per verification;
5: Generate master key K_{prp} and challenge k_{chal};
6: for vector $G^{(j)}$, j ‹- 1, n do
7: for round i ‹- 1, t do
8: Derive α_i = fk_{chal} (i) and $k_{prp}^{(i)}$ from k_{prp}
9: Compute $v^{(j)}_i = 1\alpha_i^q * G^{(j)} [\phi k_{prp}^{(i)} (q)]$
10: end for
11: end for
12: Store all the v_is locallys
13: end procedure

B. Threat Model and Security Requirements

1) 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.

2) Collusion-resistance: If multiple users collude, they may be able to decrypt a ciphertext by combining their attributes even if each of the users cannot decrypt the ciphertext alone [11]–[13]. For example, suppose there exist a user with attributes {"Battalion 1", "Region 1"} and another user with attributes {"Battalion 2", "Region 2"}. They may succeed in decrypting a ciphertext encrypted under the access policy of ("Battalion 1" AND "Region 2"), 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.

3) 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.

III. PROPOSED SCHEME

In this section, we provide a multiauthority 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 Bethencourt et al. [13], dozens of CP-ABE schemes have been proposed [7], [11]–[13]. 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. The second level key (SLK) is generated by the storage node. Commander before sending the first level encrypted message to the storage node, sends a request to the storage node for the SLK and provide the details of the users who can access the message and in return the storage node sends the key to the commander and stores the key for the users while decrypting the message. The key is generated using the token generation algorithm showed in algorithm1.

IV.CONCLUSION

Disruption-tolerant network (DTN) technologies are day by day becoming one of the successful solutions in military applications that allow wireless devices to communicate with each other and access and share the confidential information reliably by using external storage nodes. CP-ABE is a scalable cryptographic solution to the access control and secures 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 2PC protocol prevents the key authorities from obtaining any master secret information. And to get the original message, it depends on two keys, one from key authorities and another from the storage node. Thus, users are not required to fully trust the authorities in order to protect their data to be shared. The data confidentiality and privacy can be cryptographically enforced against any curious key authorities or data storage nodes in the proposed scheme.

The inherent key escrow problem is resolved such that the confidentiality of the stored data is guranteed 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.

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] 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, "Identsity-based encryption with efficient revocation," in Proc. ACM Conf. Comput. Commun. Security, 2008, pp. 417–426.