ABSTRACT

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. In this paper, we propose 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.

KEYWORDS: DTNs, OMT, CRC

Purpose

In many military network scenarios, 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 becoming successful 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 messages 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. [N. Chen, M. Gerla, D. Huang, and X. Hong, 2010]

Many military applications require increased protection of confidential data including access control methods that are cryptographically enforce. 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. [M. Chuah and P. Yang, 2006] For example, in a disruption-tolerant military network, a commander may store a 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). 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 that fulfills the requirements for secure data retrieval in DTNs.[D. Huang and M. Verma, 2009]

ABE features a mechanism that enables an access control over encrypted data using access policies and ascribed attributes among private keys and ciphertexts. Especially, cipher text-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.[J. Bethencourt, A. Sahai, and B. Waters, 2007]

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.[L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, 2003]

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.[M. M. B. Tariq, M. Ammar, and E. Zequra, 2006]

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 singlemaster 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.

Scope

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 ciphertexts 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.

1) Attribute Revocation: Bethencourt 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.

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. 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.

**Literature Survey**

**MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks:**

Disruption-tolerant networks (DTNs) attempt to route network messages via intermittently connected nodes. Routing in such environments is difficult because peers have little information about the state of the partitioned network and transfer opportunities between peers are of limited duration. In this paper, we propose MaxProp, a protocol for effective routing of DTN messages. MaxProp is based on prioritizing both the schedule of packets transmitted to other peers and the schedule of packets to be dropped. These priorities are based on the path likelihoods to peers according to historical data and also on several complementary mechanisms, including acknowledgments, a head-start for new packets, and lists of previous intermediaries. Our evaluations show that MaxProp performs better than protocols that have access to an oracle that knows the schedule of meetings between peers. Our evaluations are based on 60 days of traces from a real DTN network we have deployed on 30 buses. Our network, called UMassDieselNet, serves a large geographic area between five colleges. We also evaluate MaxProp on simulated topologies and show it performs well in a wide variety of DTN environments, networks where contemporaneous end-to-end paths are unstable or unlikely.

**Node Density-Based Adaptive Routing Scheme for Disruption Tolerant Networks:**

Traditional ad hoc routing protocols do not work in intermittently connected networks since end-to-end paths may not exist in such networks. Hence, routing mechanisms that can withstand disruptions need to be designed. A store-and-forward approach has been proposed for disruption tolerant networks. Recently, several approaches have been proposed for unicast routing in disruption-prone networks e.g. the 2-hop relay approach, delivery probability based routing, and message ferrying. In our earlier paper, we have evaluated a combined multihop and message ferrying approach in disruption tolerant networks. In that paper, we assume that a special node is designated to be a message ferry.

This makes communication possible, even when an instantaneous end-to-end path does not exist. Several routing schemes have been proposed for DTNs. They can be categorized into three categories: (i) using message ferries or data mules to connect partitioned nodes, (ii) using history-based information to estimate delivery probability of peers and pass the message to the peer that can best deliver the message, and (iii) using 2-hop relay forwarding schemes where a source can send multiple copies to different relay nodes and have the relay nodes deliver to the destination when they encounter the destination. In our earlier work, we have evaluated the performance of a multihop routing scheme with custody transfer feature in a single domain DTN.

**EXISTING SYSTEM**

Storage nodes are introduced 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. 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. We refer to this DTN architecture where multiple authorities issue and manage their own attribute keys independently as a decentralized DTN.

**PROPOSED SYSTEM**

In this project, we propose an attribute-based secure data retrieval scheme using CP-ABE for decentralized DTNs. 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. [J. Bethencourt, A. Sahai, and B. Waters, 2007]

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.

HIGH LEVEL DESIGN

The purpose of the design is to obtain the solution of a problem specified by the requirements document. This phase is the first step in moving from problem to the solution domain. In other words, starting with what is needed design takes us to work how to satisfy the needs. The design of the system is perhaps the most critical factor affecting the quality of the software and has a major impact on the later phases, particularly testing and maintenance. System design aims to identify the modules that should be in the system, the specifications of these modules and to interact with each other to produce the desired results. At the end of the system design all the major data structures, file formats, output formats as well as major modules in the system and their specifications are decided.

Design Overview

Design involves identification of classes, their relationships as well as their collaboration. In objectory, classes were divided into Entity classes, interface classes and the control classes. The Computer Aided Software Engineering tools that are available commercially do not provide any assistance in this transition. CASE tools take advantage of meta modelling are helpful only after the construction of class diagram is completed.

In the Fusion method, it used some object-oriented approaches Technique(OMT) Class Responsibility Collaborator(CRC) and Objectory, used the term like Object Modelling Agents to represent some of the hardware and software systems. In Fusion method, there was no requirement phase where in a user will supply the initial requirement document. Any software project is worked out by both analyst and designer. [S.Roy and M. Chuah, 2009]

The proposed Multi signature scheme is based on a multiparty extension of the ElGamal type signature variant: GES. The proposed Multi signature scheme can equally use any other secure and efficient signature variant of the ElGamal type signature scheme. The main reason for using the defined GES is to minimize the computational cost of generating and verifying the individual signatures and group-oriented signature in a multiparty setting without compromising security.

![System Architecture](image-url)
Project modules

1. System Model
2. Key authorities
3. Sender
4. Storage node
5. User

Modules Description

System Model

We describe the DTN architecture and define the security model. 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.[L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, 2003] 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).

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.[D. Huang and M. Verma, 2009] 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.

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.

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.

IMPLEMENTATION

Implementation of any software is always preceded by important decisions regarding selection of the platform, the language used, etc. These decisions are often influenced by several factors such as the real environment in which the system works the speed that is required, the security concerns, other implementation specific details etc.

A product software implementation method is a blueprint to get users and/or organizations running with a specific software product. The method is a set of rules and views to cope with the most common issues that occur when implementing a software product business alignment from the organizational view and acceptance from the human view. The implementation of product software, as the final link in the deployment chain of software production, is in a financial perspective of a major issue. It is stated that the implementation of (product) software consumes up to 1/3 of the budget of a software purchase (more than hardware and software requirements together).[N. Chen, M. Gerla, D. Huang, and X. Hong, 2010]

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods. Implementation is the process of converting
a new system design into operation. It is the phase that focuses on user training, site preparation and file conversion for installing a candidate system. The important factor that should be considered here is that the conversion should not disrupt the functioning of the organization. [V. Goyal, O. Pandey, A. Sahai, and B. Waters, 2006]

There are two major implementation decisions that have been made before the implementation of this project. They are as follows:

1. Selection of the platform (operating system)
2. Selection of the programming language for development of this project.

**Programming Language Selection**

Developing the Software in Java language has many advantages. Every full implementation of the Java platform gives you the following features:

- **The essentials:** Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.
- **Applets:** The set of conventions used by applets.
- **Networking:** URLs, TCP (Transmission Control Protocol), UDP (User Datagram Protocol) sockets, and IP (Internet Protocol) addresses.
- **Internationalization:** Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.
- **Security:** Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.
- **Software components:** Known as JavaBeans, can plug into existing component architectures.
- **Object serialization:** Allows lightweight persistence and communication via Remote Method Invocation (RMI).
- **Java Database Connectivity (JDBC):** Provides uniform access to a wide range of relational databases.

The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more. The following figure depicts what is in

**Code Conventions**

I have selected JAVA programming language for my project. Code conventions are important to programmers for a number of reasons. 80 percent of the lifetime cost of a piece of software goes to maintenance. Hardly any software is maintained for its whole life by the original author. Code conventions improve the readability of the software, allowing engineers to understand new code more quickly and thoroughly. If you ship your source code as a product, you need to make sure it is as well packaged and clean as any other product you create.

**Naming Conventions**

Naming conventions make programs more understandable by making them easier to read. They can also give information about the function of the identifier—for example, whether it’s a constant, package, or class—which can be helpful in understanding the code. The conventions given in this section are high level. The naming rules for the identifiers are explained below:

- **Classes:** Class names should be nouns, every class in java can be composed of following element. Field, static method, member methods, static field, constructors, parameterized types.
- **Variables:** Variable names should be short yet meaningful. The choice of a variable name should be mnemonic—that is, designed to indicate to the casual observer the intent of its use.
- **Constants:** Java does not directly support constants. However, a static final variable is effectively a constant. Java constants are normally declared in ALL CAPS. Words in java constant are normally separated by underscore. For ex:
  ```java
  Public class MaxUnits {
  Public static final int MAX_UNITS=25;
  }
  ```
- **Comments:** Comments should be used to give overviews of code and provide additional information that is not readily available in the code itself. Comments should contain only information that is relevant to reading
and understanding the program.[A. Sahai and B. Waters, 2005]

The implementation stage involves following tasks.

• Careful planning.
• Investigation of system and constraints.
• Design of methods to achieve the changeover.
• Training of the staff in the changeover phase.
• Evaluation of the changeover method.
• The method of implementation and the time scale to be adopted are found out initially. Next the system is tested properly and the same time users.

Conversion

Conversion is the process of changing from the old system to the new or modified one. Conversion should be accomplished quickly as delays and long conversion periods cause frustration and the task of all involved including the analyst and user becomes more difficult.

Conversion Plan

This plan should be formulated in consultation with the users. The conversion plan includes a description of all activities that must occur to implement the new system and put it into operation. This includes identification of people responsible and timetable for each activity that is to be carried out.[A. Lewko and B. Waters, 2009]

During the planning of conversion, the analyst should from a list containing all tasks including the following:

• List all file for conversion.
• Identity all data required to build new file conversion.
• Identity all controls to be used during conversion.
• Verify conversion schedule.

CONCLUSIONS

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. 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.

REFERENCES


D. Huang and M. Verma, “ASPE: Attribute-based secure policy enforcement in vehicular ad


