Secure Document Distribution System Using Attribute Based Encryption

In this assignment you will learn to securely share data using *Attribute-Based Encryption* (ABE) which is one of the most recent and effective techniques for accomplishing both confidentiality and access control simultaneously, and in a more flexible way than traditional encryption schemes allow. You will implement a *Secure Document Distribution System* for an organization's employees to securely share data, encrypted using attribute-based encryption, among themselves.

Attribute-based Encryption:
Attribute-based encryption is a form of public-key cryptography where the private keys of users are generated based on certain attributes they possess. The data is encrypted with an access policy that is specified in terms of the attributes of the intended recipients. Anyone possessing the attributes satisfying that policy can decrypt the data using their private key, assumed to have been generated by some trusted source. There are different variants of ABE; the one you will use in this assignment is Ciphertext-policy ABE (CP-ABE for short).

Example of CP-ABE:
For a particular file to be encrypted, the owner can specify the policy <designation: head_of_department AND department: computer_science OR department: electrical_engineering>. The ciphertext (i.e. the encrypted file) will be generated based on this policy. Only the users who are head of department in computer science and electrical engineering departments can then decrypt this file. For example, a head of department in computer science department will possess a private key (known only to him) that was generated based on his attributes. Since these attributes satisfy the policy specified by the author, he will be able to decrypt the data. This is useful because in this manner, data can be made available without knowing the identities of the recipients in advance; anyone holding the right set of attributes can decrypt the data.

Secure Document Distribution System:
You will write a simple secure document distribution system for an organization where any employee can share documents with other employees in a secure way, i.e. only those employees should be able to access the document who have the attributes to satisfy the policy that the author of the document has specified.

We will use the term "producer" for the employee who distributes the data and "consumer" for the employees who will receive the data. The producer should be able to create a data file (you may assume it is always a text file for simplicity), encrypt it specifying a policy over the attributes of the consumers who can decrypt the file, and store it. Only those consumers who have the attributes set by the publisher should be able to decrypt it. All users of the system can act as both consumers and producers.

You need to implement at least the following modules:

1. An ***Attribute Authority*** (AA) who assigns attributes to users, and maintains them in a database. Any new user who registers in the system is given attributes in the form of key value pairs. For example, say your organization is FAST; a faculty member in the Islamabad campus may be given the attributes [campus: Islamabad, designation: Assistant Professor, current courses: OOP,CNET]. “Campus” is the key and “Islamabad” is a value in this example. You can pick randomly from a predefined “attribute universe” to assign to new users.
2. A ***Key Management Authority*** (KMA) with the following responsibilities:
    a. Generate keys for new users based on their attributes and a secret that is unique to each user.
    b. Maintain a record of registered users, their attributes (assigned by AA) and their keys (which it has generated).
    c. Revoke access when required (e.g. when a producer specifies that a certain consumer or all consumers holding a specific attribute should no longer be given access to a particular document, it should be restricted immediately). Thinking of a way to implement revocation is part of your assignment. The revocation may be both attribute-level (e.g. revoke access of all users having the attribute “department: management”) or consumer-level (e.g. revoke access of a certain consumer), as requested by a producer.
3. A ***Storage Service*** (SS) that writes the generated documents to secure (usually cloud) storage and retrieves them when requested. Please note that you do not have to actually use a cloud service – just a module in your program can be a “pretend” cloud service; the files will simply be stored on your own machine. Users should be able to ask the SS to display a list of documents stored in it. Documents should be displayed as an ID and a short description, e.g. as follows:
   1111 Admission Test Instructions
   2345 Payroll System Manual
   6784 Visiting Lecturer Rates
4. An ***Encryption/Decryption Service*** that does the actual encryption and decryption actions. You should use a library instead of trying to write your own code; there are several libraries for ABE in different languages. For C++ you can use this one; do look at this example to see how CP-ABE can be easily implemented within your program.

Feel free to add other helper/administrative modules, e.g. a module to handle login and registration, another to invoke the encryption or file handling functions etc. In a real system, these modules would be running in separate servers. To emulate that, your code should have a clearly modular design, where each module only performs its own specific tasks and interacts with other modules. You are ***free to choose your own programming language***, but for example, in C++ these entities would at the least be separate classes.

In your implementation, you will come up with a sample organization to use to demonstrate your system, for example, a university, a hospital, a local government office etc. Specify an attribute universe which defines all possible attributes that can be generated within your organization, e.g. if your organization is a hospital, some of your attributes can be “department”, “designation”, “specialization” (e.g. if the employee is a doctor), “salary” etc.