**INTRODUCTION**

**Introduction to Information Forensics and Security.**

More and more people and enterprises are motivated to outsource their local document management systems to the cloud which is a promising information technique (IT) to process the explosive expanding of data. Cloud computing can collect and reorganize a huge amount of IT resources and apparently, the cloud servers can provide more secure, flexible, various, economic and personalized services compared with the local servers. Despite the advantages of cloud services, leaking the sensitive information, such as personal information, company financial data and government documents, to the public is a big threat to the data owners. In addition, to make full use of the data on the cloud, the data users need to access them flexibly and efficiently. Consequently, a huge challenge of outsourcing the data to the cloud is how to protect the confidentiality of the data properly while maintaining their searchability. An intuitive approach is encrypting the documents first and then outsourcing the encrypted documents to the cloud. A large number of searchable document encryption schemes have been proposed in the literatures, including single keyword Boolean search schemes, single keyword ranked search schemes and multi-keyword Boolean search schemes. However, all these schemes cannot support effective,flexible and efficient document search because of their simple functionalities. Privacy-preserving multi-keyword ranked document search schemes are more promising and practical. However, all the documents in these schemes are organized by a coarse-grained access control mechanism, i.e., each authorized data user can access all the encrypted documents. As an example, the whole IEEE Xplore Digital Library can be accessed by all the authorized organizations (e.g., the universities) at present and this cannot satisfy the data owners and users in the future. In this paper, a new situation is considered. A data user may want to access part of the library (e.g., computers and data related papers) and intuitively she wants to pay less money compared with the data users who want to access the whole library. In other words, in the document collection, each document can be accessed only by a set of specific data users. In this case, we need to design a fine-grained access control mechanism for the documents and it is more reasonable compared with the present method. To make the data users able to access part of IEEE Xplore Digital Library on demands, a possible approach is encrypting the documents through attribute-based encryption (ABE) schemes before outsourcing them to the cloud. Meanwhile, the authorized data users are assigned with a set of attributes. A data user can decrypt a file if and only if her attributes match the file’s attributes. Recently, ciphertext-policy attribute-based encryption (CP-ABE) is a hot research area and it can provide fine-grained, one to many and flexible access control. In these schemes, each document is encrypted individually and their encryption efficiency can be improved by employing hierarchical attribute-ba sed encryption schemes. However, these schemes cannot be employed directly to solve our problem properly. First, most existing schemes focus on encrypting a single access tree. However, it is impossible that all the documents in IEEE Xplore Digital Library share a single access tree and how to construct a set of optimized access trees for the document collection is a huge challenge. Second, in most existing schemes, when the documents are mapped to a set of shared access trees, the data users need to store a large number of secret keys which will be analyzed. Apparently, this is a heavy burden for the data users especially for an extremely large document collection and how to decrease the amount of secret keys for the data users is another challenge. Except for access control, document search efficiency is also a challenge for a large document collection. To our knowledge, most existing schemes cannot support timeefficient retrieval over the documents which are organized under attribute-based access control m echanism. To support the previously discussed service, we first design an algorithm to generate hierarchical access trees for the document collection. The proposed algorithm employs the greedy strategy to build the access trees incrementally and each access tree grows by continuously splitting the nodes in the tree. Then we design a ciphertext-policy attribute-based hierarchical document collection encryption scheme called CP-ABHE. In the proposed scheme, a set of documents can share a same integrated access tree and be encrypted together rather than being encrypted individually. In this way, both the ciphertext storage space and time costs of the encryption/decryption are saved. The security of the proposed scheme is proved theoretically and its effectiveness is also evaluated by simulation.

To support accurate and efficient document search over the encrypted documents, a complicated index structure is then constructed for the document collection. We first map the documents to document vectors based on the TF-IDF model and, in addition, the attributes of the documents are also taken into consideration. The similarity function between the document vectors is carefully designed and the vectors are organized based on their relative similarities in the ARF tree. Specifically, the similar vectors compose micro clusters which are then aggregated with each other to generate macro clusters until all the vectors belong to one cluster. The ARF vectors of the nodes in the tree are used to describe the inherent properties of the clusters represented by the nodes. At last, a depth-first search algorithm for the ARF tree is designed to guarantee both the search efficiency and accuracy. The main contributions of this paper are summarized as follows:

• A practical hierarchical attribute-based document collection encryption scheme is proposed in which the documents are organized and controlled based on attributes. The proposed scheme can greatly decrease the storage and computing burdens.

• We map the documents to vectors in which both the keywords and associated attributes are considered. The ARF tree is proposed to organize the document vectors and support time-efficient document retrieval. In addition, a depth-first search algorithm is designed.

• A thorough simulation is performed to illustrate the security, efficiency and effectiveness of our scheme. Specifically, the proposed encryption scheme performs very well in both time and storage efficiency. In addition, our scheme also provides efficient and accurate document retrieval method.

In this free Data Science tutorial you will have the introduction to Data Scientist roles and responsibilities, machine learning algorithms,data analysis, data manipulation, data frame, random forest, linear and logistic regression, decision trees, neural networks, Java language, Java libraries, data model, variable, set, and more.There are plenty of Data Science use cases and practical examples.Data science helps the user by providing an ability to analyze huge data sets and by doing necessary operations, data science will save precious time and makes some big profit out of it.

**Description**

Data science is very much popular in today’s world scenario as there is a huge amount of data generated each day in different fields like mart, hospitals, colleges, etc.

Users need to perform some operations by analyzing the dataset and then find something useful from that data

**Data Science Process**

**Step 1: Organize Data**

It includes the physical storage and formatting of data and integrated finest practices in data management

**Step2: Package Data**

In this the prototypes are created, the visualization is built and also statistics is performed. It includes logically joining and manipulating the raw data into a new representation and package**.**

**Step 3: Deliver Data**

In this process data is delivered to those who need that data.

Data is the new Oil. This statement shows how every modern IT system is driven by capturing, storing and analysing data for various needs. Be it about making decision for business, forecasting weather, studying protein structures in biology or designing a marketing campaign. All of these scenarios involve a multidisciplinary approach of using mathematical models, statistics, graphs, databases and of course the business or scientific logic behind the data analysis. So we need a programming language which can cater to all these diverse needs of data science. Java shines bright as one such language as it has numerous libraries and built in features which makes it easy to tackle the needs of Data science.

In this tutorial we will cover these the various techniques used in data science using the Java programming language.

Data science is the process of deriving knowledge and insights from a huge and diverse set of data through organizing, processing and analysing the data. It involves many different disciplines like mathematical and statistical modelling, extracting data from it source and applying data visualization techniques. Often it also involves handling big data technologies to gather both structured and unstructured data. Below we will see some example scenarios where Data science is used.

Recommendation systems

As online shopping becomes more prevalent, the e-commerce platforms are able to capture users shopping preferences as well as the performance of various products in the market. This leads to creation of recommendation systems which create models predicting the shoppers needs and show the products the shopper is most likely to buy.

Financial Risk management

The financial risk involving loans and credits are better analysed by using the customers past spend habits, past defaults, other financial commitments and many socio-economic indicators. These data is gathered from various sources in different formats. Organising them together and getting insight into customers profile needs the help of Data science. The outcome is minimizing loss for the financial organization by avoiding bad debt.

Improvement in Health Care services

The health care industry deals with a variety of data which can be classified into technical data, financial data, patient information, drug information and legal rules. All this data need to be analysed in a coordinated manner to produce insights that will save cost both for the health care provider and care receiver while remaining legally compliant.

Computer Vision

The advancement in recognizing an image by a computer involves processing large sets of image data from multiple objects of same category. For example, Face recognition. These data sets are modelled, and algorithms are created to apply the model to newer images to get a satisfactory result. Processing of these huge data sets and creation of models need various tools used in Data science.

Efficient Management of Energy

As the demand for energy consumption soars, the energy producing companies need to manage the various phases of the energy production and distribution more efficiently. This involves optimizing the production methods, the storage and distribution mechanisms as well as studying the customers consumption patterns. Linking the data from all these sources and deriving insight seems a daunting task. This is made easier by using the tools of data science.

Java in Data Science

The programming requirements of data science demands a very versatile yet flexible language which is simple to write the code but can handle highly complex mathematical processing. Java is most suited for such requirements as it has already established itself both as a language for general computing as well as scientific computing. More over it is being continuously upgraded in form of new addition to its plethora of libraries aimed at different programming requirements. Below we will discuss such features of java which makes it the preferred language for data science.

A simple and easy to learn language which achieves result in fewer lines of code than other similar languages like java. Its simplicity also makes it robust to handle complex scenarios with minimal code and much less confusion on the general flow of the program.

It is cross platform, so the same code works in multiple environments without needing any change. That makes it perfect to be used in a multi-environment setup easily.

It executes faster than other similar languages used for data analysis like R and Its excellent memory management capability, especially garbage collection makes it versatile in gracefully managing very large volume of data transformation, slicing, dicing and visualization.

Most importantly Java has got a very large collection of libraries which serve as special purpose analysis tools. For example – the NumPy package deals with scientific computing and its array needs much less memory than the conventional java list for managing numeric data. And the number of such packages is continuously growing.

Java has packages which can directly use the code from other languages like Java or C. This helps in optimizing the code performance by using existing code of other languages, whenever it gives a better result.

In the subsequent chapters we will see how we can leverage these features of java to accomplish all the tasks needed in the different areas of Data Science.