Secure *k*-NN Query on Encrypted Cloud Data with Multiple Keys

ABSTRACT

Cloud computing has become an increasingly popular service for its flexibility and scalability, which motivates many organizations, institutions and companies to prefer to outsource data services to cloud platform. At the same time, much attention has been paid to cope with the special security and privacy problems in outsourced cloud. On one hand, to protect the data confidentiality, the data owner (DO) encrypt the sensitive information of his outsourced data, such as income level, health records, personal photos before the dataset is uploaded to the cloud. On the other hand, data owner may plan to rely on cloud platform for querying of the datasets stored in cloud, not just for storage and management. Therefore, a large amount of secure schemes have been proposed to support the query over encrypted cloud data.

EXISTING SYSTEM

As a fundamental query operation in spatial and multimedia databases, *k*-nearest neighbors (*k*-NN) query aims at identifying *k* nearest points for a given query point in a dataset. In the past few years, researchers have proposed various methods to address the security and privacy problems of *k*-NN query on encrypted cloud data. The general approach is to encrypt data by the data owner (DO) before outsourcing; the authorized query users (QUs) perform a complex series of encryption and decryption operations during query execution. For example, the work in [8] proposes an asymmetric scalar-product-preserving encryption (ASPE) to preserve scalar product between the query vector and any vector for distance comparison, which is sufficient to find *k*-NN. Instead of finding exact nearest neighbor, Yao et al. [9] allow a cloud party to approximate it based on secure Voronoi diagram algorithm. Elmehdwi et al. [10] propose a novel protocol over encrypted data based on a Twin-Cloud [11] model and Paillier cryptosystem [12], which can calculate *k*-NN between data records and query records in a secure manner. However, all the above schemes have assumed that the query users are fully-trusted and have the access to the key for encrypting and decrypting outsourced data. It will bring about several problems in the real world.

DRAWBACK

- Cloud platform can totally break the outsourced database once the key is obtained from any compromised query user. It is obvious that each query user could be one of the lucrative targets for attackers.

- Data owner may have no enough trust on each query user in many applications which will limit the scope of these schemes.

- Once query users receive the key, their query processing will not be controlled by data owner any more, and it is difficult to revoke the access even they are deemed to be untrustworthy. In general, these schemes with key-sharing are still far from being practical in most instances.

PROPOSED SYSTEM

We propose a novel scheme for secure *k*-NN query on encrypted cloud data with multiple keys, in which the DO and each QU all hold their own different keys, and do not share them with each other; meanwhile, the DO encrypts and decrypts outsourced data using the key of his own. Our scheme is constructed by a distributed two trapdoors public-key cryptosystem (DT-PKC) and a set of protocols of secure two-party computation, which not only preserves the data confidentiality and query privacy but also supports the offline data owner.

ADVANTAGES

- Secure *k*-NN query on encrypted cloud data with multiple keys

- Distributed two trapdoors public-key cryptosystem (DT-PKC) [18], we construct a set of protocols of secure two-party computation that will be used as sub-routines of our proposed scheme.

SYSTEM REQUIREMENTS

- **H/W System Configuration:-**

➢ Processor	-	Pentium –IV
➢ RAM	-	4 GB (min)
➢ Hard Disk	-	20 GB
➢ Key Board	-	Standard Windows Keyboard
➢ Mouse	-	Two or Three Button Mouse
➢ Monitor	-	SVGA

- **S/W System Configuration:-**

➢ Operating System	:	Windows 7 or 8 32 bit
➢ Application Server	:	Tomcat5.0/6.X
➢ Backend coding	:	Java
➢ Tool	:	Virtual Box

- Environment : Ubuntu
- Technology : Hadoop