# Secure Sensitive Data Sharing on A Big Data Platform

**Srisail.S. kanajagi**
Asst prof
Sharanbasva university

***Abstract-*** *Vast amount of user's sensitive data are stored on the big data platform. The sharing of sensitive data will help enterprises to reduce the cost of providing users with personalized service, and achieve value added services of data. However, secure data sharing is problematic. Security is one of the most difficult task to implement in any environment. Different forms of attacks in the application side and in the hardware components. By analyzing the present security situation of sensitive data, will propose a framework for secure sharing of those data on big data platform, including security submission, storage, use and destruction of sensitive data on the semi-trusted big data platform.*

*Relevant key technologies were studied, such as the proxy re-encryption algorithm based on heterogeneous cipher-text transformation and user process protection methods based on the virtual machine monitor, which provides support for the realization of system functions. The framework well protects the security of user's sensitive data, and shares these data effectively and safely. At the same time the data owners have complete control of their data, which is conducive to foster a sound environment for modern Internet information security.*

## I. INTRODUCTION

1.1 What Is Big Data

Big records is records that exceeds the processing potential of traditional database systems.

The statistics is just too big, actions too fast, or does not match the systems of traditional databasearchitectures. In other words, Big records is an all-encompassing time period for any series of information units so large and complex that it becomes tough to technique using on-hand information management tools or conventional information processing applications. To advantage cost from this information, you must choose an alternative manner to technique it. Big Data is the next technology of statistics warehousing and business analytics and is poised to exponential growth and availability of information, both dependent and unstructured.

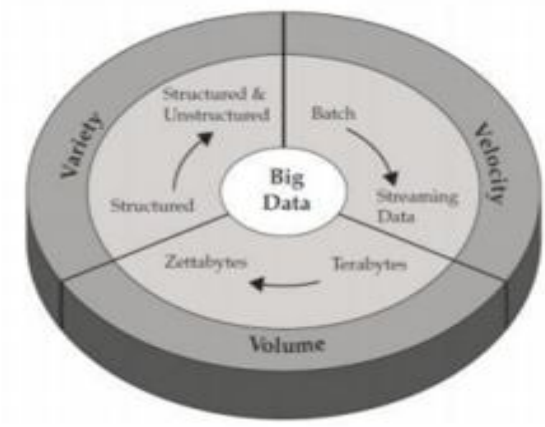
Every day, we create 2.5 quintillion bytes of facts — so much that 90% of the data within the world today has been created within the remaining years alone. This information comes from everywhere: sensors used to collect weather information, posts to social media sites, digital images and videos, buy transaction records, and cell cellphone GPS signals to name a few. This facts is big information.

### 1.2 Definition Of Big Data

Big data usually includes data sets with sizes beyond the ability of commonly used software tools to capture, create, manage, and process the data within a tolerable elapsed time. Big data is high-volume, high-velocity and high-variety information assets that demand costeffective, innovative forms of information processing for enhanced insight and decision-making.

### 1.3 Characteristics Of Big Data

Big Data is a term that describes the large volume of data – both structured and unstructured – that inundates a business on a day-to-day basis. But it's not the amount of data that's important. It's what organizations do with the data that matters. Big data can be analyzed for insights that lead to better decisions and strategic business moves. Doug Laney articulated the now-mainstream definition of big datadeliver top line sales cost effectively for enterprises. Bigrecords is a famous term used to describe the and information from sensor or machine-to-machine data. In the past, storing it would've been a problem – but new technologies (such as Hadoop) have eased the burden.



Velocity: Data streams in at an unprecedented speed and must be dealt with in a timely manner. RFID tags, sensors and smart metering are driving the need to deal with torrents of data in nearreal time.

Variety: Data comes in all types of formats – from structured, numeric data in traditional databases to unstructured text documents, email, video, audio, stock ticker data and financial transactions.

## II. LITERATURE SERVEY

### 2.1 Fine-Grained Data Access Control Systems With User Accountability In Cloud Computing

For the purpose of helping the data owner impose fine-grained access control of data stored on untrusted cloud servers, a feasible solution would be encrypting data through certain cryptographic primitive(s). Goyal V, Pandey O, SahaiA,Waters B propose a new cryptosystem for fine-grained sharing of encrypted data that called Key-Policy Attribute-Based Encryption (KP-ABE). This scheme uses a set of attributes to describe the encrypted data and builds a access policy in user's private key. If attributes of the encrypted data can satisfy the access structure in user's private key, then the user can obtain the message through decrypt algorithm. In a key-policy attribute based encryption (KP-ABE) system, ciphertexts are labeled by the sender with a set of descriptive attributes, while user's private key is issued by the trusted attribute authority captures an policy which is also called the access structure.KP-ABE schemes are suitable for structured organizations with rules about who may read particular documents.

### 2.2 A Secure Self-Destructing Scheme For Electronic Data

As more and more services and applications are emerging in the Internet, exposing user sensitive data in the Internet becomes more easily. The simplest way to protect the security of sensitive user data is to encrypt the data in advance, and then disclose the data as the three Vs:

Volume: Organizations collect data from a variety of sources, including business transactions, social media decryption key is exposed to unauthorized users. In this paper, we propose a secure selfdestructing scheme for electronic data (SSDD for short). We achieve this goal by first encrypting the data, and then distributing both the decryption key and a part of the ciphertext into the distributed hash table (DHT) network.

### 2.3 Overshadow: A Virtualization-Based Approach To Retrofitting Protection In Commodity Operating Systems.

Commodity operating systems entrusted with securing sensitive data are remarkably large and complex, and consequently, frequently prone to compromise. To address this limitation, we introduce a virtual-machine-based system called Overshadow that protects the privacy and integrity of application data, even in the event of a total OS compromise. Overshadow presents an application with a normal view of its resources, but the OS with an encrypted view. This allows the operating system to carry out the complex task of managing an application's resources, without allowing it to read or modify them. Thus, Overshadow offers a last line of defense for application data. Overshadow builds on multi-shadowing, a novel mechanism that presents different views of "physical" memory, depending on the context performing the access. This primitive offers an additional dimension of protection beyond the hierarchical protection domains implemented by traditional operating systems and processor architectures.

### 2.4 Processing Private Queries Over Untrusted Data Cloud Through Privacy Homomorphism.

Query processing that preserves both the data privacy of the owner and the query privacy of the client is a new research problem. It shows increasing importance as cloud computing drives more businesses to outsource their data and querying services. However, most existing studies, including those on data outsourcing, address the data privacy and query privacy separately and cannot be applied to this problem. In this paper, we propose a holistic and efficient solution that comprises a secure traversal framework and an encryption scheme based on privacy homomorphism. decryption key only to those authorized users. However, the sensitive user data will be leaked while the Ntypical queries such as k-nearest-neighbor queries (kNN) on R-tree index.

### 2.5 Identity-Based Proxy Re-Encryption

An Identity-Based Proxy Re-encryption (IB-PRE) scheme is an extended Identity Based Encryption scheme. The first extension is an algorithm that generates re-encryption keys that can be given to the proxy. The proxy uses the second algorithm to apply these re-encryption keys to ciphertexts and "atomically" re-encrypt them from one identity to another. In a non-interactive scheme, re-encryption keys may be generated by the delegator using only her IBE secret key the IBE master secret is not required.

## III. PROBLEM STATEMENT

### 3.1 Existing System

Users store vast amounts of sensitive data on a big data platform. Sharing sensitive data will help enterprises reduce the cost of providing users with personalized services and provide value-added data services. However, secure data sharing is problematic. With the rapid development of information digitization, massive amounts of structured, semi-structured, and unstructured data are generated quickly. By collecting, sorting, analyzing, and mining these data, an enterprise can obtain large amounts of individual users' sensitive data. Regarding encryption technology, the Attribute-Based Encryption (ABE) algorithm includes Key-Policy ABE (KPABE)[1] and Ciphertext-Policy ABE (CPABE)[2]. ABE decryption rules are contained in the encryption algorithm, avoiding the costs of frequent key distribution in ciphertext access control. However, when the access control strategy changes dynamically, a data owner is required to reencrypt the data.

### 3.1.1 Disadvantages

- ABE decryption rules are contained in the encryption algorithm
- when the access control strategy changes dynamically,a data owner is required to reencrypt the data.
- A semi-trusted agent with a proxy key can re-
- The framework is scalable to large datasets by leveraging an indexbased approach. Based on this framework, we devise secure protocols for processing
- Storage (ABACCS), Each user's private key is labeled with a set of attributes, and data is encrypted with an attribute condition restricting the user to be able to decrypt the data only if their attributes satisfy the data's condition.
- The problem of how to prevent sensitive information from leaking, when an emerge ncy occurs.
- The open source cloud computing storage system, Hadoop Distributed File System (HDFS), cannot destroy data completely, which may lead to data leak.

### 3.2 Proposed System

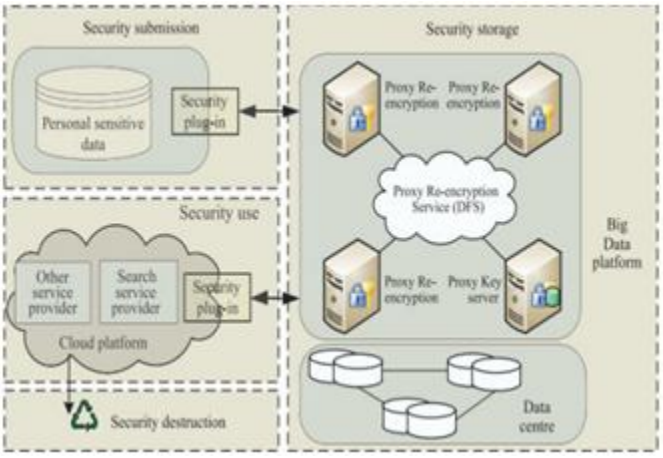
Systematic framework for secure sensitive data sharing on a big data platform including reliable submission, safe storage, riskless use, and secure destruction on a semi trusted big data sharing platform. We present a Proxy Re-Encryption based on heterogeneous ciphertext transformation and a user protection method based on a virtual machine monitor, which provides support for the realization of system functions. The framework protects the security of user's sensitive data effectively and shares these data safely. A common and popular method of ensuring data submission security on a semi-trusted big data platform is to encrypt data before submitting data to the platform. Some operations (such as encryption, decryption, and authorization) are provided using a security plug-in. A cloud platform service provider (such as an SESP) using data on a big data platform ensures data security by downloading and using the security plug-in.

### 3.2.1 Advantages

- Reduces the exposure of sensitive data.
- Simplifies security auditing and testing.
- Enables automated security management.
- Improves redundancy and disaster recovery.
- Achieve compliance with leading self-regulatory frameworks.
- Identify and authenticate users beforegranting access.
- Prevent and curtail external attacks. encrypt ciphertext.
- Attribute-Based Access Control for Cloud
- Encrypt sensitive and confidential information assets whenever feasible

## IV. SYSTEM DESIGN

Issuing and renting sensitive data on a semitrusted big data platform requires a data security mechanism. Building secure channels for a full sensitive data life cycle requires consideration of four aspects of safety problems : Reliable submission, Safe storage, Riskless use, Secure destruction. The below figure 4.1 shows systematic framework for secure sensitive data sharing on a big data platform.




The basic flow of the framework is as follows. First, enterprises that have individual users' sensitive information pre-set those service providers that need to share this sensitive information and then submit and store the corresponding encrypted data on a big data platform using the local security plugin. Second, we need to perform the required operation with the submitted data using PRE on the big data platform.

Then, cloud platform service providers who want to share the sensitive information download and decrypt the corresponding data in the private process space using the secure plug-in with sensitive privacy data running in that space. Last, we use a secure mechanism to destroy used data still stored temporarily in the cloud. In short, the framework protects the security of the full sensitive data life cycle effectively. Meanwhile, data owners have complete control over their own data. Next, we discuss the most critical PRE algorithm based on heterogeneous cipher-text transformation and user process protection methods using the VMM.

4.1 Secure Submission and Storage of Sensitive Data Based on PRE

Data submission and storage (sharing) operations:

- The browser reads the data files uploaded by the data owner, generates randomly an AES transparent encryption key (Symmetric Encryption Key, SEK), and then use the AES algorithm to encrypt the data files;
- Uses the PRE algorithm to encrypt the SEK and store the data ciphertext and SEK ciphertext in the data centers;
- Identifies from the data owner the users designated to share the data
- Uses the security plug-in to read the private key of the data owner and obtain the data user's public key from the big data platform;
- Uses the security plug-in to generate the corresponding PRE key using the EncIBE function and to upload the PRE key to the authorization key server of the big data platform
- Re-encrypts the data using the ReEnc function on the big data platform, thereby generating PRE ciphertext.

Data extraction operations

- The browser queries whether there is authorization for the data user on the PRE server of the big data platform, and if an authorization is in effect, proceeds to Step (2)
- Uses the download plug-ins to send data download requests to the big data platform, which then finds PRE ciphertext data in the data center
- Pushes the PRE ciphertext to the secure data plug-in on the big data platform
- Invokes a data user's download plug-in to read the user's private key and prepares to decrypt data
- Invokes a data user's download plug-in to decrypt received SEK ciphertext using the DecPKE function and obtain the AES symmetric key
- Permits the data user to decrypt the data ciphertext using the AES symmetric key to

## V. IMPLEMENTATION

### 5.1 Modules

Issuing and renting sensitive data on a semitrusted big data platform requires a data security mechanism. Building secure channels for a full sensitive data life cycle requires consideration of four aspects of safety problems

- Reliable submission
- Safe storage
- Riskless use
- Secure destruction.

### 5.2 Modules Description

#### 5.2.1 Reliable Submission

A common and popular method of ensuring data submission security on a semi-trusted big data platform is to encrypt data before submitting data to the platform. Some operations (such as encryption, decryption, and authorization) are provided using a security plug-in. A cloud platform service provider (such as an SESP) using data on a big data platform ensures data security by downloading and using the security plug-in.

#### 5.2.2 Safe Storage

To ensure secure storage, we make use of Heterogeneous Proxy Re-Encryption (H-PRE), which supports heterogeneous transformation from IdentityBased Encryption (IBE) to PublicKey Encryption (PKE). H-PRE is compatible with traditional cryptography. The intent is to transform the cipher data that the owner uploads into ciphertext that the data user can decrypt using his or her own private key.

#### 5.2.3 Riskless Use

We assume that the cloud cannot be trusted, and that the decrypted clear text will leak users' private information. Therefore, we need to adopt VMM, through a trusted VMM layer, bypassing the guest operating system and providing data protection directly to the user process. The key management module of the VMM is used for storing public keys of the new register program group. When a program is running, the symmetric key at the bottom of the main program will be decrypted dynamically by the key management module. All applications of the public and symmetric keys are stored in the

memory of the VMM. process protection technology based on a obtain the required clear text.

**5.2.4 Secure Destruction**

The archive, replication, and backup mechanism of cloud storage create data redundancy, requiring the use of a suitable data destruction scheme to delete the user's private personal data. To achieve high security, we designed a lease-based mechanism to destroy private data and keys thoroughly in a controlled manner. Clear text and keys exist nowhere in the cloud, after the lease expires.

**5.3 Algorithm : Heterogeneous Proxy Re-Encryption (H-PRE).**

Step 1: SetupIBE(k)

Input security parameters k, generate randomly a primary security parameter mk, calculate the system parameter set params using a bilinear map and hash function.

Step 2 : KeyGenIBE(mk, params, id)

When the user requests the private key from the key generation center, the key generation center obtains the legal identity (id) of the user and generates the public and private keys (pkid, skid) for the user using params and mk.

Step 3 : KeyGenPKE(params)

When a user submits a request, the key management center not only generates the identity-based public and private keys, but also generates the public and private keys of the traditional public key system (pk0 id, sk0 id).

Step 4 : EncIBE(pkid; skid; params;m)

When the user encrypts data, the data owner encrypts the clear-text (m) into the ciphertext (c =(c1.c2) using the user's own (pkid, skid) and a random number (r 2 RZp *).

Step 5 : KeyGenRE(skidi; sk0 idi' pk id', params)

When the data owner (user i) grants user j permissions, using skidi , sk0 idi , and pk0 idj , user i computes the PRE key completing the transformation from user i to user j .

Step 6 : ReEnc(ci,rkidi-rkidj,params)

This process is executed transparently on the big data platform. The function re-encrypts the ciphertext that user i encrypted into ciphertext that user j can decrypt. It inputs c(ci= (ci1.ci2)), the PRE key(rkidi-rkidj), and related system parameters, and

Step 7 : DecPKE(cj',sk'idj,params)

This is a function for decrypting the PRE ciphertext. After receiving the PRE ciphertext (cj=(cj1.cj2)) from the proxy server of the big data platform, user j determines the clear-text (m'=m) of the data using his or her own sk0 idj.

## VI. CONCLUSION

In summary, proposed a systematic framework of secure sharing of sensitive data on big data platform, which ensures secure submission and storage of sensitive data based on the heterogeneous proxy re-encryption algorithm, and guarantees secure use of clear text in the cloud platform by the private space of user process based on the VMM. The proposed framework well protects the security of users' sensitive data. At the same time the data owners have the complete control of their own data, which is a feasible solution to balance the benefits of involved parties under the semi-trusted conditions. In the future, will optimize the heterogeneous proxy reencryption algorithm, and further improve the efficiency of encryption. In addition, reducing the overhead of the interaction among involved parties is also an important future work.

## REFERENCES

[1] J. Li, G. Zhao, X. Chen, D. Xie, C. Rong, W. Li, L. Tang, and Y. Tang," Fine-grained data access control systems with user accountability in cloud computing".
[2] L. Wang, L. Wang, M. Mambo, and E. Okamoto, "New identity-based proxy re encryption schemes to prevent collusion attacks".
[3] H. Hu, J. Xu, C. Ren, and B. Choi, "Processing private queries over untrusted data cloud through privacy homomorphism".
[4] C. Hong, M. Zhang, and D. Feng, "AB-ACCS: A cryptographic access control scheme for cloud storage"
[5] X. Chen, T. Garfinkel, E. C. Lewis, and B. Spasojevic, "Overshadow: A virtualizationbased approach to retrofitting protection in commodity operating systems".
[6] G. Wang, F. Yue, and Q. Liu, "A secure self-destructing scheme for electronic data".then the big data platform computes and outputs the PRE ciphertext (cj=(cj1.cj2)).
[7] S.Ruj, M.Stojmenoovic andA.Nayak "privacy preserving access control with authentication for securing data in cloud".

[8] S.SeenuIropia,R.VijayLaxmi, ”Decentralized Access Control Of Data Stored In Cloud Using Key-Policy Attribute Based Encryption”.
[9] Elisa Bertino, Beng Chin Ooi, Yanjiang Yang, and Robert H. Deng. “Privacy and ownership preserving of outsourced medical data”.
[10] Trusted Computing Group, TNC architecture for interoperability, http://www.trustedcomputinggroup.org/ resources/tnc architecture for interoperability specification.
[11] L. Zeng, Z. Shi, S. Xu, and D. Feng, “Safevanish: An improved data self-destruction for protecting data privacy”.