International Journal of Scientific Engineering and Technology
Volume No.5 Issue No.5, pp: 228-231

ISSN:2277-1581
01 May 2016

Assured Sensitive Data Sharing on a Big Data Platform using Proxy Re-Encryption (PRE) Algorithm

K.Sudhakar Y.Rama mohan P.N.V.S Pavan Kumar S.Vinay Kumar
CSE dept, G. Pulla Reddy Engineering College , Kurnool.
sudhakarcs14@gmail.com

Abstract: 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. This paper proposes a framework for secure sensitive data sharing on a big data platform, including secure data delivery, storage, big data sharing platform. We present a proxy re-encryption algorithm based on heterogeneous cipher text transformation The framework protects the security of users' sensitive data effectively and shares these data safely.

Key words: secure sharing, sensitive data, big data, proxy re-encryption;

1 Introduction

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. These data not only meet the demands of the enterprise itself, but also provide services to other businesses if the data are stored on a big data platform. Traditional cloud storage merely stores plain text or encrypted data passively. Such data can be considered as "dead", because they are not involved in calculation. However, a big data platform allows the exchange of data (including sensitive data). It provides mass data storage and computational services. Computation services refer primarily to operations (such as encrypting data, conversion, or function encryption) on data used by participants, which can invigorate "dead" data. An example of such an application is shown in Fig. 1 to illustrate the flow process of sensitive data on such a platform

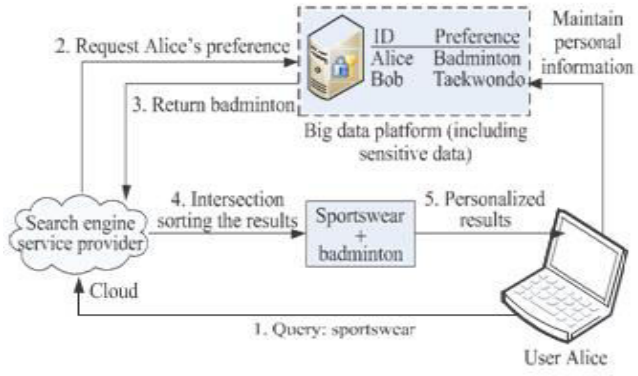

Fig. 1 Application of sensitive data (user's preferences).

In Fig. 1, we consider user's preferences as sensitive data. When Alice submits a query (sportswear), the Search Engine Service Provider (SESP) first looks for Alice's preference on the big data platform. If the big data platform has collected and shared the user's personal preference information, "badminton", then the search engine returns personalized results (sportswear + badminton) to Alice. When Alice sees her favorite badminton sportswear, she experiences a pleasant purchase. Consequently, this leads to a win- win situation. However, while data sharing increases enterprise assets, Internet insecurity and the potential of sensitive data leakage also create security issues for sensitive data sharing.

Secure sensitive data sharing involves four primary safety factors. First, there are security issues when sensitive data are transmitted from a data owner's local server to a big data platform. Second, there can be sensitive data computing and storage security problems on the big data platform. Third, there are secure sensitive data use issues on the cloud platform. Fourth, there are issues involving secure data destruction. Some research institutions and scholars at home and abroad have made positive contributions to exploration and research aimed at solving these security problems.

Existing technologies have partially resolved data sharing and privacy protection issues from various perspectives, but they have not considered the entire process in the full data security life cycle. However, a big data platform is a complete system with multi- stakeholder involvement, and thus cannot tolerate any security breach resulting in sensitive data loss. In this paper, we describe a system model created to ensure secure sensitive data sharing on a big data platform, to guarantee secure storage on the big data platform using Proxy Re-Encryption (PRE) technology.

The rest of this paper is organized as follows. Section 2 describes related work. Section 3 proposes a systematic framework for secure sensitive data sharing. Section 4 explains secure submission and storage of sensitive data based on PRE in detail.

Conclusions are drawn in Section 6.

2 Related Work

In this section, we focus on relevant topic such as encryption, access control, in a cloud storage environment.

2.1 Data encryption and access control of cloud storage

ISSN:2277-1581
01 May 2016

Regarding encryption technology, the Attribute-Based Encryption (ABE) algorithm includes Key-Policy ABE (KP-ABE)[1] and Cipher text-Policy ABE (CP- ABE)[2] . ABE decryption rules are contained in the encryption algorithm, avoiding the costs of frequent key distribution in cipher text access control. However, when the access control strategy changes dynamically, a data owner is required to re-encrypt the data. A method based on PRE is proposed in Ref. [3]. A semi-trusted agent with a proxy key can re-encrypt cipher text; however, the agent cannot obtain the corresponding plaintext or compute the decryption key of either party in the authorization process[4] . A Fully Homomorphic Encryption (FHE) mechanism is proposed in Ref. [5]. The FHE mechanism permits a specific algebraic operation based on cipher text that yields a still encrypted result. More specifically, retrieval and comparison of the encrypted data produce correct results, but the data are not decrypted throughout the entire process. The FHE scheme requires very substantial computation, and it is not always easy to implement with existing technology. In regard to cipher text retrieval with a view toward data privacy protection in the cloud, cipher text retrieval solutions in the cloud are proposed in Refs. [6–8]. Regarding access control, a new cryptographic access control scheme, Attribute-Based Access Control for Cloud Storage (AB-ACCS), is proposed in Ref. [9]. 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. Distributed systems with Information Flow Control (DIFC)[10] use a tag to track data based on a set of simple data tracking rules. DIFC allow untrusted software to use private data, but use trusted code to control whether the private data can be revealed. In Ref. [11], the authors consider the complexity of fine-grained access control for a large number of users in the cloud and propose a secure and efficient revocation scheme based on a modified CP-ABE algorithm. This algorithm is used to establish fine-grained access control in which users are revoked according to Shamir's theory of secret sharing. With a Single Sign-On (SSO), any authorized user can log in to the cloud storage system using a standard common application interface

3.Systematic Framework for Secure Sensitive Data Sharing on a Big Data Platform

Issuing and renting sensitive data on a semi- trusted 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, and secure destruction. A systematic framework for secure sensitive data sharing on a big data platform is shown in Fig. 2.

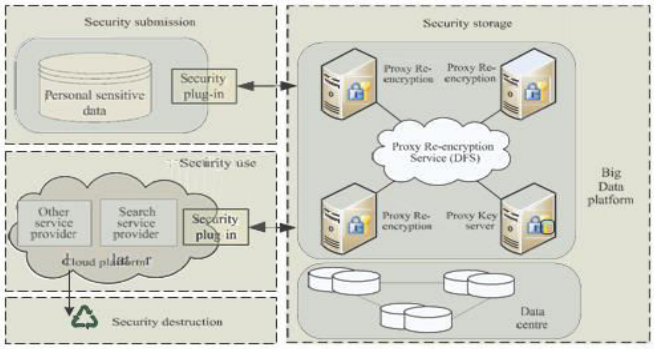

Fig. 2 Systematic framework for secure sensitive data sharing on a big data platform

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.

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

We assume that the cloud cannot be trusted, and that the decrypted clear text will leak users' private information. Therefore, we need to adopt process protection technology based on a 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.

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.

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 plug- in. 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

4 Secure Submission and Storage of Sensitive Data Based on PRE

4.1 H-PRE

H-PRE involves three types of algorithm, traditional identity-based encryption (including $Setup_{IBE}$, $KeyGen_{IBE}$, Enc_{IBE}, and Dec_{IBE}), re-encryption (including $KeyGen_{RE}$, ReEnc, and ReDec functions), and the last one is the traditional public key cryptosystems (including $KeyGen_{PKE}$, Enc_{PKE}, and Dec_{PKE}). The basic H-PRE process is simple. The data owner encrypts sensitive data using a local security plug-in and then uploads the encrypted data to a big data platform. The data are transformed into the cipher text that can be decrypted by a specified user after PRE services. If an SESP is the specified user, then the SESP can decrypt the data using its own private key to obtain the corresponding clear text. We complete the following steps to implement the H-PRE algorithm.

(1) $Setup_{IBE}$.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.

(2) $KeyGen_{IBE}$.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 (pk_{id}, sk_{id}) for the user using params and mk.

(3) $KeyGen_{PKE}$.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.and transforms the original cipher using the PRE key. Then, PRE ciphertext, which can be encrypted by the (authorized) data users, is generated. If the data user wants to use the data on the big data platform, the data user will send data requests to the platform and then query whether there is corresponding data in the shared space. If such data exist, the data user accesses and downloads it. The operation on the big data platform is independent and transparent to users. Moreover, the computing resources of the big data platform are more powerful than those of the client. Hence, we can put PRE computational overhead on the big data platform to improve user experience. The PRE system includes data submission, storage (sharing), and data extraction

(4) Enc_{IBE_i}.pk_{id} ; skid ; params; m/: When the user encrypts data, the data owner encrypts the clear-text (m) into the ciphertext (c D .c_1 ; c_2 /) using the user's own (pk_{id}, sk_{id}) and a random number (r 2 RZ).operations.Data submission and storage (sharing) operations After receiving the data uploading request, the Web browser invokes the security plug-in and provides data(5) $KeyGen_{RE}$.$skid_{\mathbf{i}}$; $sk_{id_{\mathbf{i}}}$, sk^0; params/: When the uploading services for the data owner, in accordance data owner (user i) grants user **j** permissions, using with the following detailed steps. The browser (1) $sk_{id_{\mathbf{i}}}$, sk^0, and pk^0 , user i computes the PRE key reads the data files uploaded by the data owner, ($rkid_{\mathbf{i}}$ $id_{\mathbf{j}}$), completing the transformation from user I to user **j** .

(6) ReEnc.c_i ; $rk_{id_{\mathbf{i}}}$ $id_{\mathbf{j}}$; params/: This process is executed transparently on the big data platform. The function re-encrypts the cipher text that user i encrypted into cipher text that user **j** can decrypt. It inputs c_i .c_i D .c_{i1} ; c_{i2} //, the PRE key ($rk_{id_{\mathbf{i}}}$ $id_{\mathbf{j}}$), and related system parameters, and then the big data platform computes and outputs the PRE cipher text ($c_{\mathbf{j}}$ D .$c_{\mathbf{j}1}$; $c_{\mathbf{j}}$ 2 /). generates randomly an AES transparent encryption key (Symmetric Encryption Key, SEK), and then use the AES algorithm to encrypt the data files; (2) uses the PRE algorithm to encrypt the SEK and store the data cipher text and SEK cipher text in the data centers; (3) identifies from the data owner the users designated to share the data; (4) 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; (5) uses (7) Dec_{PKE}.$c_{\mathbf{j}}$; sk^0

; params/: This is a function for the security plug-in to generate the corresponding PRE decrypting the PRE cipher text. After receiving the PRE cipher text ($c_{\mathbf{j}}$ D .$c_{\mathbf{j}1}$; $c_{\mathbf{j}}$ 2 /) from the proxy server of the big data platform, user **j** determines the clear-text (m^0 D m) of the data using his or her own sk^0 .

4.2The submission, storage, and extraction operations of system sensitive data

The data owner encrypts data locally, first using the Advanced Encryption Standard (AES) symmetric encryption algorithm to encrypt the submission data and then using the PRE algorithm to encrypt the symmetric key of the data. These results are all stored within the distributed data. In the meantime, if the data owner shares the sensitive data with other users, the data owner must authorize the sensitive data locally and generate the PRE key, which is stored in the authorization key server.

On the big data platform, the PRE server re-encrypts key using the EncIBE function and to upload the PRE key to the authorization key server of the big data platform; and (6) re-encrypts the data using the ReEnc function on the big data platform, thereby generating PRE cipher text.

Data extraction operations After receiving the data download request, the Web browser invokes the security plug-in and provides data download services for the data user, in accordance with the following detailed steps. The browser (1) 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); (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; (3) pushes the PRE ciphertext to the secure data plug-in on the big data platform; (4) invokes a datauser's download plug-into read the user's private key and prepares to

ISSN:2277-1581
01 May 2016

decrypt data; (5) invokes a data user's download plug-in to decrypt received SEK ciphertext using the DecPKE function and obtain the AES symmetric key; and (6) permits the data user to decrypt the data ciphertext using the AES symmetric key to obtain the required clear text.

The data extraction operation is put into the private space of a user process by the secure plug-in, a prerequisite for secure use of sensitive data.

6 Conclusions

In summary, we 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, we will optimize the heterogeneous proxy re-encryption 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

i. S. Yu, C. Wang, K. Ren, and W. Lou, Attribute based data sharing with attribute revocation, in Proc. 5th ACM Symposium on Information, Computer and Communications Security, Beijing, China, 2010, pp. 261– 270.

ii. J. Bethencourt, A. Sahai, and B. Waters, Ciphertext-policy attribute-based encryption, in Proc. IEEE Symposium on Security and Privacy, Oakland, USA, 2007, pp. 321–334.

iii. 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, in Proc. 2nd Int. Conf. on Cloud Computing, Indianapolis, USA, 2010, pp. 89–96.

iv. L. Wang, L. Wang, M. Mambo, and E. Okamoto, New identity-based proxy re-encryption schemes to prevent collusion attacks, in Proc. 4th Int. Conf. Pairing-Based Cryptograghy-Pairing, Ishikawa, Japan, 2010, pp. 327–346.

v. C. Gentry, A fully homorphic encryption scheme, Ph.D dissertation, Stanford University, California, USA, 2009.

vi. S. Ananthi, M.S. Sendil, and S. Karthik, Privacy preserving keyword search over encrypted cloud data, in Proc. 1st Advances in Computing and Communications, Kochi, India, 2011, pp. 480–487.

vii. H. Hu, J. Xu, C. Ren, and B. Choi, Processing private queries over untrusted data cloud through privacy homomorphism, in Proc. 27th IEEE Int. Conf. on Data Engineering, Hannover, Germany, 2011, pp. 601–612.

viii. N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, Privacy-preserving multi-keyword ranked search over encrypted cloud data, in Proc. 30th IEEE INFOCOM, Shanghai, China, 2011, pp. 829–837.

ix. C. Hong, M. Zhang, and D. Feng, AB-ACCS: A cryptographic access control scheme for cloud storage, (in Chinese), Journal of Computer Research and Development, vol. 47, no. 1, pp. 259–265, 2010.

x. N. Zeldovich, S. Boyd-Wickizer, and D. Mazieres, Securing distributed systems with information flow control, in Proc. 5th USENIX Symposium on Networked Systems Design and Implementation, San Francisco, USA,2008, pp. 293–308.

xi. Z. Lv, C. Hong, M. Zhang, and D. Feng, A secure and efficient revocation scheme for fine-grained access control in cloud storage, in Proc. 4th IEEE Int. Conf. on Cloud Computing Technology and Science, Taipei, Taiwan, China, 2012, pp. 545–550.

xii. A. M. Azab, P. Ning, E. C. Sezer, and X. Zhang, HIMA: A hypervisor-based integrity measurement agent, in Proc. 25th Annual Computer Security Applications Conf., Hawaii, USA, 2009, pp. 461–470.

xiii. A. M. Azab, P. Ning, Z. Wang, X. Jiang, X. Zhang, and N. C. Skalsky, HyperSentry: Enabling stealthy in-context measurement of hypervisor integrity, in Proc. 17th ACM Conference on Computer and Communications Security, Chicago, USA, 2010, pp. 38–49.

xiv. Trusted Computing_ Group, TNC architecture for interoperability, http://www.trustedcomputinggroup.org/resources/tnc architecture for interoperability specification, 2014.

xv. H. Zhang, L. Chen, and L. Zhang, Research on trusted network connection, (in Chinese), Chinese Journal of Computers, vol. 33, no. 4, pp. 706–717, 2010.

xvi. D. Feng, Y. Qin, D. Wang, and X. Chu, Research on trusted computing technology, (in Chinese), Journal of ComputerResearch and Development, vol. 48, no. 8, pp. 1332–1349, 2011.

xvii. F. Zhang, J. Chen, H. Chen, and B. Zang, Cloudvisor: Retrofitting protection of virtual machines in multi-tenant cloud with nested virtualization, in Proc. 23rd ACM Symposium on Operating Systems Principles, Cascais, Portugal, 2011, pp. 203–216.

K. Sudhakar
Assistant professor of cse dept
G Pulla Reddy Engineering college
Kurnool
sudhakarcs14@gmail.com

Y. Rama Mohan
Assistant professor of cse dept
G Pulla Reddy Engineering college
Kurnool
yrmgprec@gmail.com

P. N.V.S. Pavan Kumar
Assistant professor
CSE DEPT
G Pulla Reddy Engineering college
Kurnool
pegatraj@gmail.com

S. Vinay Kumar
Assistant professor
CSE DEPT
G Pulla Reddy Engineering college
Kurnool
vinay.gprec@gmail.com