IFERP
connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

An Efficient and Fine-grained Big Data Access Control Scheme with Privacy preserving Policy

[1] M. Leela Amani, [2] PK. VenkateswarLal, [3] M. Sujatha, [4] D. samatha, [5] G. Subhasri
[1][2][3][4][5] Computer science & Engineering, Narayana Engineering College, Gudur.

Abstract: How to manage the access of the massive amount of huge knowledge becomes a awfully difficult issue, particularly once huge knowledge ar keep within thecloud. Ciphertext-Policy Attribute based Encryption (CP-ABE) may be a promising secret writing technique that {allows} end-users to encode their knowledge beneath the access policies outlined over some attributes {of knowledge |of knowledge| of information} shoppers and solely allows data shoppers whose attributes satisfy the access policies to decode the information. In CP-ABE, the access policy is connected to the ciphertext in plaintext type, which can also leak some non-public data regarding end-users. Existing ways solely part hide the attribute values within the access policies, whereas the attribute names ar still unprotected. During this paper, we have a tendency to propose AN economical and fine-grained huge knowledge access management theme with privacy-preserving policy. Specifically, we have a tendency to hide the complete attribute (rather than solely its values) within the access policies. to help knowledge coding, we have a tendency to additionally style a unique Attribute Bloom Filter to judge whether or not AN attribute is within the access policy and find the precise position within the access policy if it's within the accesspolicy. Security analysis and performance analysis show thatour theme will preserve the privacy from any LSSS accesspolicy while not using a lot of overhead.

Index Terms—Big Data; Access Control; Privacy-preserving Policy; Attribute Bloom Filter; LSSS Access Structure

INTRODUCTION

In the era of huge knowledge, a large quantity of knowledge will be generated quickly from varied sources (e.g., good phones, sensors, machines, social networks, etc.). Towards these huge knowledge, standard pc systems don't seem to be competent to store and method these knowledge. Thanks to the versatile and elastic computing resources, cloud computing may be a natural suited storing and process huge knowledge [1], [2]. With cloud computing, end-users store their knowledge into the cloud, and have faith in the cloud server to share their knowledge to alternative users (data consumers). so as to solely share end-users' knowledge to approved users, it's necessary to style access management mechanisms in keeping with the necessities of end-users. Once outsourcing knowledge into the cloud, end-users lose the physical management of their knowledge. Moreover, cloud serviceproviders don't seem to be fully-trusted by end-users, that makes the access management tougher. Parenthetically, if the normal access management mechanisms (e.g., Access management Lists) ar applied, the cloud server becomes the decide to judge the access policy and build access call. Thus, end-users could worry that the cloud server could build wrong access call advisedly or accidentally, and disclose their knowledge to some unauthorized users. So as to alter end-users to manage the access of their own knowledge, some attribute-based access management schemes are planned by investing attribute-based secret writing. In attribute based mostly access management, end-users first outline access policies for his or her knowledge and encode the information beneath these access policies. Solely the users whose attributes will satisfy the access policy are eligible to decode the information. Though the present attribute-based access management schemes will touch upon the attribute revocation drawback, all of them suffer from one problem: the access policy may leak privacy. This is often as a result of the access policy is related to the encrypted knowledge in plaintext type. From the plaintext of access policy, the adversaries could get some privacy data regarding the end-user. parenthetically, Alice encrypts her knowledge to alter the "Psychology Doctor" to access. So, the access policy could contain the attributes "Psychology" and "Doctor". If anyone sees this knowledge, though he/she might not be ready to decode the information, he/she still will guess that Alice could suffer from some psychological issues, that leaks the privacy of Alice. to forestall the privacy escape from the access policy, an easy methodology is to cover the attributes within the access policy. However, once the attributes are hidden, not solely the unauthorized users however additionally the approved users cannot understand that attributes are concerned within the access policy, that makes the coding a difficult drawback. Thanks to this reason, existing ways don't hide or anonymize the attributes. Instead, they solely hide the values of every attribute by victimisation wildcards,

Hidden Vector secret writing and scalar product secret writing. Activity the values of attributes will somehow defend user privacy, however the attribute name might also leak non-public data. Moreover, most of those part hidden policy schemes solely support specific policy structures (e.g., AND-gates on multivalent attributes). During this paper, we have a tendency to aim to cover the complete attribute instead of solely part activity the attribute values. Moreover, we have a tendency to don't limit our methodology to some specific access structures. the fundamental plan is to precise the access policy in LSSS access structure (M;r) wherever M may be a policy matrix ANd r matcheseach row Mi of the matrix M to an attribute, and conceal the attributes by merely removing the attribute matching functionr. While not the attribute matching operate r, it's necessary to style AN attribute localization rule to judge whether or not AN attribute is within the access policy and if therefore notice the right position within the access policy. to the present finish, we have a tendency to any build anovel Attribute Bloom Filter to find the attributes to the anonymous access policy, which might save heaps of storage overhead and computation value particularly for giant attribute universe. Our contributions are summarized as follows.
1) We have a tendency to propose AN economical and fine-gained huge knowledge access management theme with privacy-preserving policy, wherever the complete attributes ar hidden within the access policy instead of solely the values of the attributes.
2)We have a tendency to additionally style a unique Attribute Bloom Filter to judge whether or not AN attribute is within the access policy and find the precise position within the access policy if it's within the access policy. 3) we have a tendency to any offer the safety proof and performance analysis of our planned theme, that demonstrate that our theme will preserve the privacy from any LSSS access policy while not using a lot of overhead. the rest of this paper is organized as follows. we have a tendency to initial describe the connected add Section. II. In Section III, We introduce some preliminary data. Section IV initial defines the system model, and then defines our theme and its security model. The elaborated construction of our theme is represented in Section V. Section VI provides the safety analysis and performance analysis of our theme. Finally, the conclusion is drawn in Section VII.

II. CONNECTED WORK

In order to alter end-users to manage the access of their own knowledge keep on untrusted remote servers (e.g., cloud servers), encryption-based access management is an efficient methodology, wherever knowledge ar encrypted by end-users and solely approved users are given coding keys. This may additionally forestall the information security throughout the transmission over wireless networks that are prone to several threats. However, ancient public key secret writing ways don't seem to be appropriate for encoding as a result of it's going to turn out multiple copies of cipher text for an equivalent knowledge once there ar several knowledge shoppers within the system. So as to affect this issue, some attribute based mostly access management schemes are planned by investing attribute-based secret writing, that solely produces one copy of cipher text for every knowledge and doesn't have to be compelled to knowledge several supposed knowledge shoppers throughout the information secret writing. Moreover, once the cloud knowledge ar encrypted, some searchable secret writing algorithms, are planned to support search on encrypted cloud knowledge.

REFERENCES

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," [Recommendations of the National Institute of Standards and Technology- Special Publication 800-145], 2011.

[2] R. Lu, H. Zhu, X. Liu, J. K. Liu, and J. Shao, "Toward efficient and privacy-preserving computing in big data era," IEEE Network, vol. 28, no. 4, pp. 46–50, 2014.

[3] K. Yang and X. Jia, "Expressive, efficient, and revocable data access control for multi-authority cloud storage," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 7, pp. 1735–1744, July 2014.