KeyD: Secure Key-Deduplication with Identity-Based Broadcast Encryption

[1]Mrs. S. Abirami, [2]Nidhin R, [3]Balakrishnan P, [4]Naveen R

[1]Assistant Professor, [2,3,4]Students
Department of Computer Science and Engineering,
Agni College of Technology, Thalambur.

***Abstract*: Deduplication, which can save storage cost by enabling us to store only one copy of identical data, becomes unprecedentedly significant with the dramatic increase in data stored in the cloud. For the purpose of ensuring data confidentiality, they are usually encrypted before outsourced. Traditional encryption will inevitably result in multiple different ciphertexts produced from the same plaintext by different users' secret keys, which hinders data deduplication. Convergent encryption makes deduplication possible since it naturally encrypts the same plaintexts into the same ciphertexts. One attendant problem is how to reliably and effectively manage a huge number of convergent keys. Several deduplication schemes have been proposed to deal with the convergent key management problem. However, they either need to introduce key management servers or require interaction between data owners. In this paper, we design a novel client-side deduplication protocol named KeyD without such an independent key management server by utilizing the identity-based broadcast encryption (IBBE) technique. Users only interact with the cloud service provider (CSP) during the process of data upload and download. Security analysis demonstrates that KeyD ensures data confidentiality and convergent key security, and well protects the ownership privacy simultaneously. A thorough and detailed performance comparison shows that our scheme makes a better tradeoff among the storage cost, communication and computation overhead.**

***Keywords*: Data deduplication, convergent encryption, convergent key management, and identity-based broadcast encryption.**

2. INTRODUCTION

THE stored data is growing intensely with the advent of the era of Big Data. We need to constantly increase the storage devices if we continue using the traditional storage way. Alternatively, more and more users are prone to outsource their storage to cloud such as Amazon Web Services (AWS) for economic savings. The ever-increasing data and users, coupled with multiple backup and other factors, result in more and more duplication of files or blocks in the cloud. In order to improve the storage efficiency in the pays-you-go model, deduplication operation is adopted for eliminating duplicate copies of redundant data on the cloud side. Consider an example that m users outsource the same data copies1 of n TB to the CSP. With data deduplication, only one copy is actually stored in the cloud, and the subsequent instances are referenced back to the saved copy for reducing storage roughly from Mn to n TB. However, in order to protect the safety of the outsourced data, they are usually encrypted by their owners before outsourced to the CSP. Then it comes the problem, how can the CSP perform deduplication when these same data copies are encrypted into different ciphertexts by different users.

3. PROBLEM STATEMENT

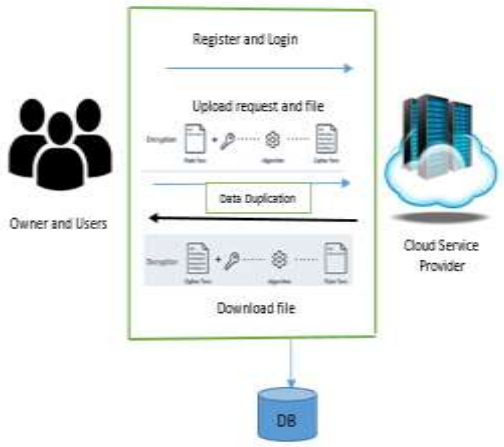
In the Existing System Deduplication, which can save storage cost by enabling us to store only one copy of identical data, becomes unprecedentedly significant with the dramatic increase in data stored in the cloud. For the purpose of ensuring data confidentiality, they are usually encrypted before outsourced. Traditional encryption will inevitably result in multiple different ciphertexts produced from the same plaintext by different users' secret keys, which hinders data deduplication. Convergent encryption makes deduplication possible since it naturally encrypts the same plaintexts into the same ciphertexts. One attendant problem is how to reliably and effectively manage a huge number of convergent keys. Several deduplication schemes have been proposed to deal with the convergent key management problem. However, they either need to introduce key management servers or require interaction between data owners. Our work falls in the category of client-side deduplication and can work in both file-level and block level, without the aid of key management servers or other trusted third parties. So there is no security for the data in the cloud.

4. PROPOSED SYSTEM

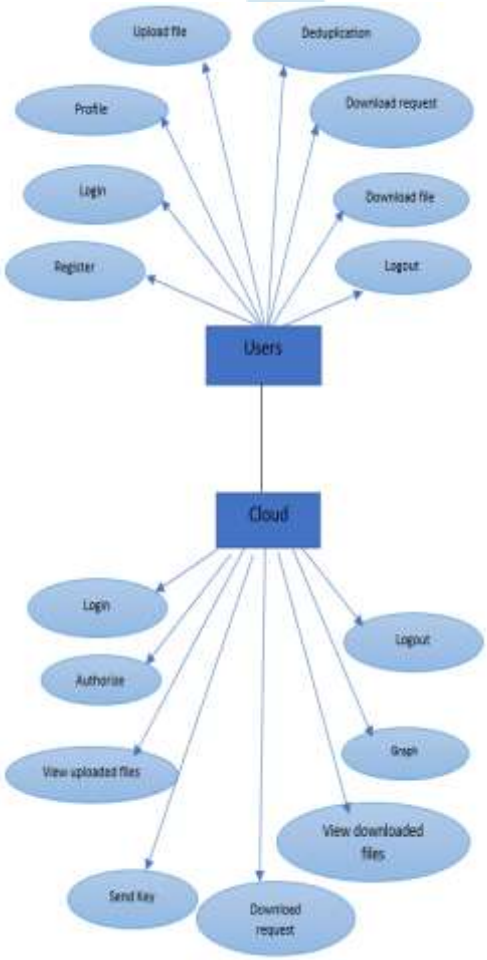
In this paper, we design a novel client-side deduplication protocol named KeyD without such an independent key management server by utilizing the identity-based broadcast encryption (IBBE) technique. Users only interact with the cloud service provider (CSP) during the process of data upload and download. Security analysis demonstrates that KeyD ensures data confidentiality and convergent key security, and well protects the ownership privacy simultaneously. A thorough and detailed performance comparison shows that our scheme makes a better tradeoff among the storage cost, communication and computation overhead. The security analysis shows that our KeyD ensures the confidentiality of data and security of convergent keys, and well protects the user ownership privacy at the same time.The security goal of our scheme is to realize data confidentiality, semantic

security of convergent keys and ownership privacy. We propose a novel client-side deduplication scheme. Specifically, we make a combination of convergent encryption (CE) and ID-based broadcast encryption (IBBE) to achieve secure and efficient convergent key management, without introducing any other independent key management servers or trusted third parties. • Security analysis demonstrates that our scheme ensures the confidentiality of data files and the security of convergent keys. • A comprehensive performance comparison between KeyD and several present works is given, showing that our .scheme makes a better tradeoff among the storage cost, communication overhead and computation overhead

5. SYSTEM ARCHITECTURE



ER Diagram



6. REQUIREMENT ANALYSIS:

Software Environment

Java Technology

❖ Java technology is both a programming language and a platform.

The Java Programming Language

❖ The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Architecture neutral Interpreted
- Object oriented Multithreaded
- Portable Robust
- Distributed Dynamic
- High performance Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called *Java byte codes* —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works. You can think of Java byte codes as the machine code instructions for the *Java Virtual Machine* (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make "write once, run anywhere" possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

7. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the system meets its requirements and user expectations and does not fail in an unacceptable manner. are various types of test.

8. APPLICATION AND FUTURE ENHANCEMENT

This application should be used for all cloud uploaded files in that website. For our future work, we will try to seek ways to protect the identity privacy of data owners, which is not considered in our scheme.

9. CONCLUSION

In this paper, we propose a secure client-side deduplication scheme KeyD to effectively manage convergent keys. Data deduplication in our design is achieved by interactions between data owners and the Cloud Service Provider (CSP), without participation of other trusted third parties or Key Management Cloud Service Providers. The security analysis shows that our KeyD ensures the confidentiality of data and security of convergent keys, and well protects the user ownership privacy at the same time. Experimental results demonstrate that the security of our scheme is not at the expense of the performance.

REFERENCES

1. Amazon Web Services, [Online]. Available: https://aws.amazon.com/cn/.
2. D.A. Sarma, X. Dong, and A. Halevy, Bootstrapping pay-as-you-go data integration systems[C]. ACM SIGMOD International
3. Conference on Management of Data, SIGMOD 2008, Vancouver, Bc, Canada, June. DBLP, 2008:861-874.
4. J.R. Douceur, A. Adya, W.J. Bolosky, D. Simon, and M. Theimer, Reclaiming Space from Duplicate Files in a Serverless Distributed File System[C]. Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on. IEEE, 2002: 617-624.
5. S. Ghemawat, H. Gobioff, and S. Leung, The Google File System[M]. SOSP '03 Proceedings of the nineteenth ACM symposium on Operating systems principles, 2003, 37(5): 29-43.
6. D. Borthakur, HDFS architecture guide[J]. Hadoop Apache Project, 2008, 53.

7. J. Li, X. Chen, M. Li, J. Li, P.P.C. Lee, and W. Lou, Secure Deduplication with Efficient and Reliable Convergent Key Management [J]. IEEE transactions on parallel and distributed systems, 2014, 25(6): 16151625.
8. G.R. Blakley and C.A. Meadows, Security of Ramp Schemes[C]. Crypto. 1984, 84: 242-268.
9. A.D. Santis and B. Masucci, Multiple Ramp Schemes [J]. IEEE Transactions on Information Theory, 1999, 45(5): 1720-1728.
10. M. Wen, K. Ota, H. Li, J. Lei, C. Gu, and Z. Su, Secure Data Deduplication with Reliable Key Management for Dynamic Updates in Cpss[J]. IEEE transactions on computational social systems, 2015, 2(4): 137-147.