Secure Key Deduplication with Identity Based Broadcast Encryption

T. Mounish[1] Mrs. I. Madhavi Latha[2]
[1]Student [2]Assistant Professor
[1,2]Department of Computer Applications
[1,2]KMM Institute of PG Studies, Tirupati, India

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 cipher texts 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 cipher texts. 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.

Key words: Data Deduplication, Convergent Encryption, Convergent Key Management, Identity-Based Broadcast Encryption

I. 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 out- source 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 pay- as-you-go model [1], reduplication operation is adopted for 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. A naive solution to this problem is described in. Each data owner encrypts all his data copies using the corresponding convergent keys and further encrypts these convergent keys using his master key. Both encrypted data copies and convergent keys are stored in the cloud, and users keep their master keys and the metadata about the outsourced data locally. Although the encrypted data can be deduplicated by the cloud, the storage of encrypted convergent keys will increase linearly with the number of users.

We propose a novel client-side deduplication scheme. Specifically, we make a combination of convergent encryption (CE) [3] and ID-based broadcast encryption (IBBE) to achieve secure and efficient convergent key management, without introducing any other independent key management servers or trust- ed 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.

II. Relative study

A. Secure Deduplication of Encrypted information while not further freelance Servers

Encrypting information on client-side before uploading it to cloud storage is crucial for shielding users' privacy but client-side encoding is at odds with the quality apply of deduplication. accommodative client-side encoding with cross-user deduplication is a vigorous analysis topic. We have a tendency to gift the primary secure cross-user deduplication theme that supports client-side encoding while not requiring any extra freelance servers. Curiously, the theme is predicated on employing a PAKE (password attested key exchange) protocol. We have a tendency to demonstrate that our theme provides higher security guarantees than previous efforts. We have a tendency to show each the effectiveness and therefore the potency of our theme, via simulations mistreatment realistic datasets Associate in Nursingd an implementation.

B. A Secure information Deduplication theme for Cloud Storage

As a lot of company and personal users source their information to cloud storage suppliers, recent information breach incidents build end-toend coding associate progressively outstanding demand. Sadly, semantically secure coding schemes render varied efficient storage optimisation techniques, like information deduplication[7], ineffective. We have a tendency to gift a completely unique concept differentiates information in line with their quality. supported this concept, we have a tendency to style associate coding theme that guarantees linguistics security for less-travelled information and provides weaker security and higher storage and information measure edges for well-liked information.

This way, information deduplication may be effective for well-liked information, while semantically secure coding protects less-traveled content. We have a tendency to show that our theme is secure underneath the radially symmetrical External Decisional Diffie-Hellman Assumption within the random oracle model. Server assisted coding for Deduplicated Storage Cloud storage service suppliers like Dropbox, Mozy, et al perform deduplication to avoid wasting house by solely storing one copy of every file uploaded. ought to shoppers conventionally cipher their files, however, savings area unit lost. Message-locked coding (the most outstanding manifestation of that is focussed encryption) resolves this tension but it's inherently subject to brute-force attacks which will recover files falling into a renowned set. we have a tendency to propose associate design that has secure deduplicated storage resisting brute-force attacks, and know it in an exceedingly system known as DupLESS. In DupLESS, shoppers cipher underneath message-based keys obtained from a key-server via associate oblivious PRF protocol.

It allows shoppers to store encrypted information with associate existing service, have the service perform deduplication on their behalf, and however achieves sturdy confidentiality guarantees. We have a tendency to show that coding for deduplicated storage can do performance and house savings about to that of mistreatment the storage service with plaintext information.

III. Proposed Algoritham

Several deduplication schemes have been proposed to deal with the convergent key management problem[6]. Proposed two secure deduplication schemes in single-server storage and distributed storage systems, while independent metadata servers needed to store metadata in the later. Anderson et al. designed a deduplication scheme for fast and secure laptop backups, which successfully reduced the number of files to be scanned and hence decreased back- up times. A local server is needed to implement multi-user authentication to shared data proposed the notion of proof of ownership (PoW), which enables the client to prove to the server that he has a copy of the file, without actually sending the file. It's essentially an interactive protocol performed between a prover (user) and a verifier (server).

A. *Application Architecture:*

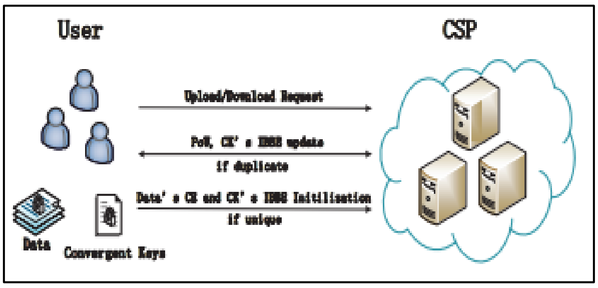


Fig. 1: The System Model of KeyD

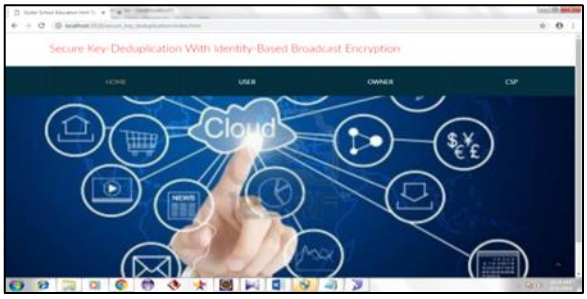
IV. Algorithm

A. Identity-based Broadcast Encryption

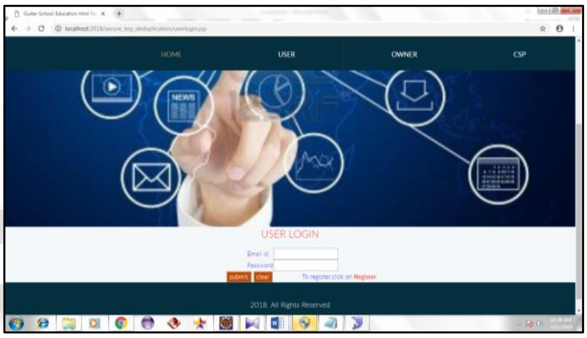
We adopt the Identity-Based Broadcast Encryption (IBBE) to encrypt convergent keys before sending them to the Cloud Service Provider (CSP). A necessary authority involved in an IBBE is the Private Key Generator PKG. Using its master secret key MSK; the PKG can generate a decryption key skIDi for each new member with identity IDi to decrypt messages. An attractive feature of the IBBE scheme is that the broadcaster does not hold any private information. Messages can be encrypted with the help of a public key PK and the set S of identities of the receivers. Then all the identities in S are able to decrypt the messages.

V. Results

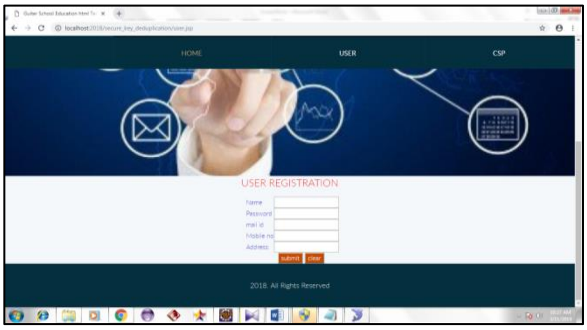
A. *Homepage*


Secure Key-Deduplication With Identity-Based Broadcast Encryption
HOME
USER
OWNER
CSP
Cloud

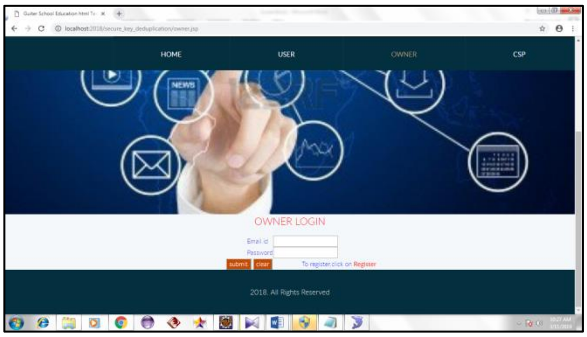
B. *User Login*



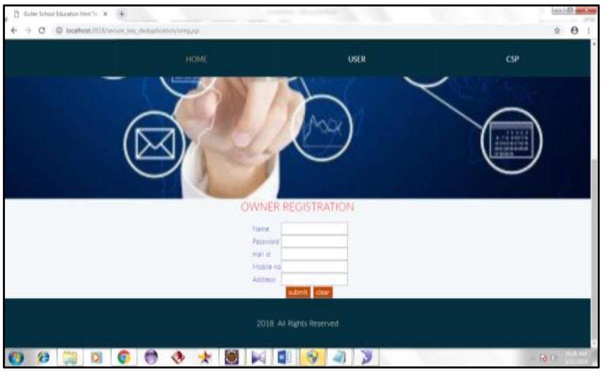
C. *User Registration*



D. *Owner Login*



E. Owner Registration

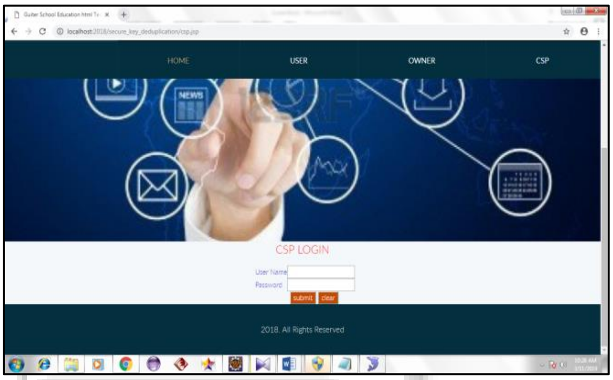


OWNER REGISTRATION

Name
Password
mail id
Mobile no
Address

2018. All Rights Reserved

F. CSP Login

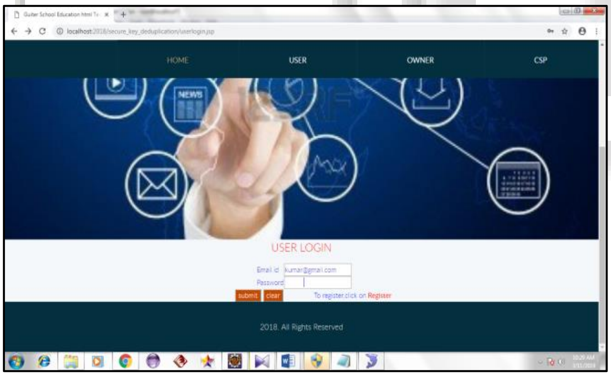


CSP LOGIN

User Name
Password

2018. All Rights Reserved

G. User Login



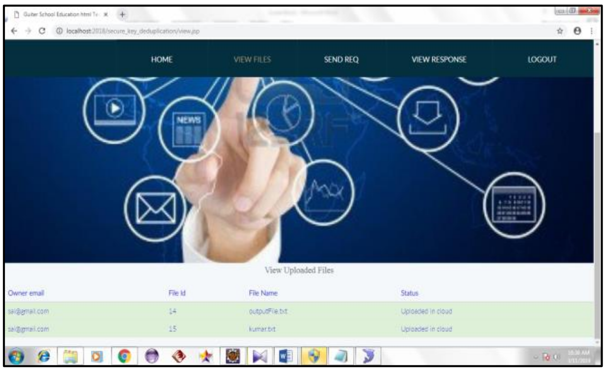
USER LOGIN

Email id kumar@gmail.com
Password

To register click on Register

2018. All Rights Reserved

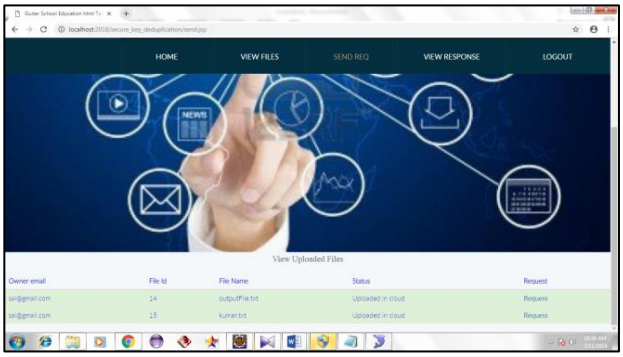
H. View user Files



View Uploaded Files

Owner email	File Id	File Name	Status
sai@gmail.com	14	outputFile.txt	Uploaded in cloud
sai@gmail.com	15	kumar.txt	Uploaded in cloud

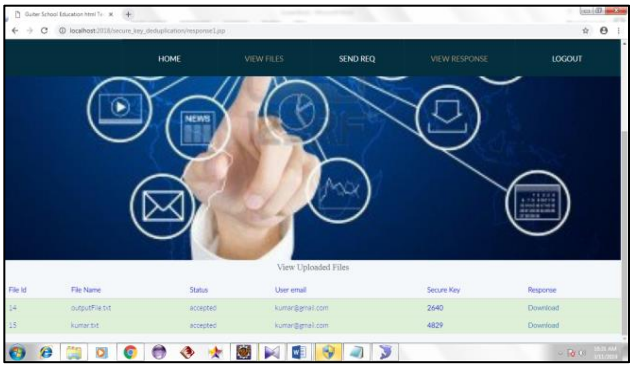
I. User Send Request



View Uploaded Files

Owner email	File Id	File Name	Status	Request
sai@gmail.com	14	outputFile.txt	Uploaded in cloud	Request
sai@gmail.com	15	kumar.txt	Uploaded in cloud	Request

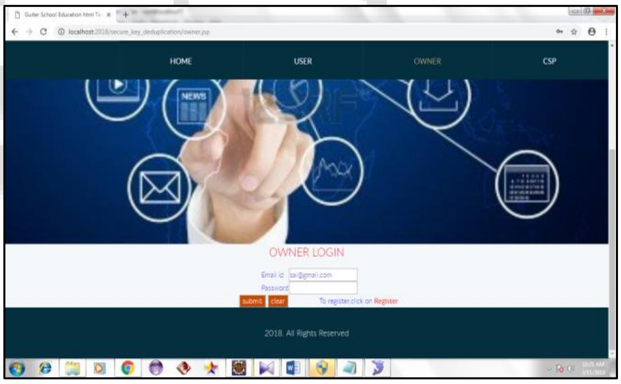
J. User View Response



View Uploaded Files

File Id	File Name	Status	User email	Secure Key	Response
14	outputFile.txt	accepted	kumar@gmail.com	2640	Download
15	kumar.txt	accepted	kumar@gmail.com	4829	Download

K. Owner Login



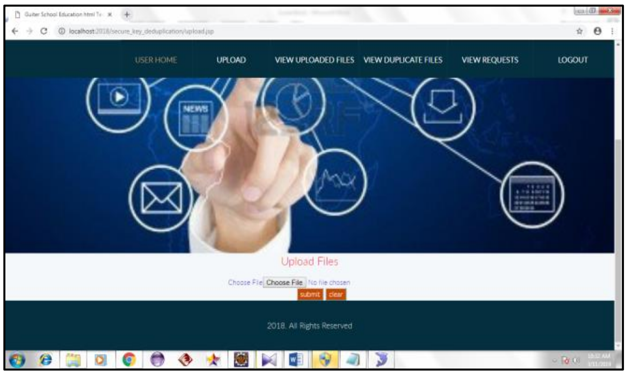
OWNER LOGIN

Email id sai@gmail.com
Password

To register click on Register

2018. All Rights Reserved

L. Owner Upload Files

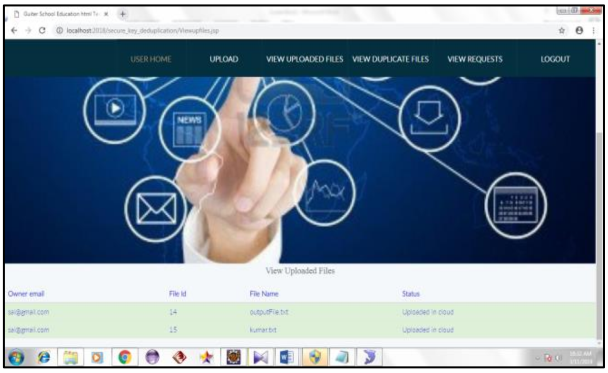


Upload Files

Choose File No file chosen

2018. All Rights Reserved

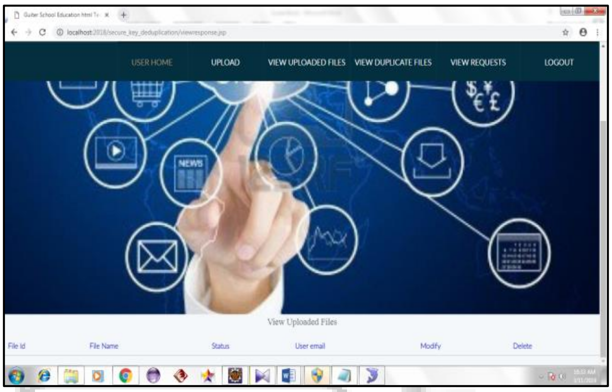
M. Owner View Upload Files


Guiter School Education html Te
localhost:2018/secure_key_deduplication/Viewupfiles.jsp
USER HOME
UPLOAD
VIEW UPLOADED FILES
VIEW DUPLICATE FILES
VIEW REQUESTS
LOGOUT
NEWS

View Uploaded Files

Owner email	File Id	File Name	Status
sai@gmail.com	14	outputFile.txt	Uploaded in cloud
sai@gmail.com	15	kumar.txt	Uploaded in cloud

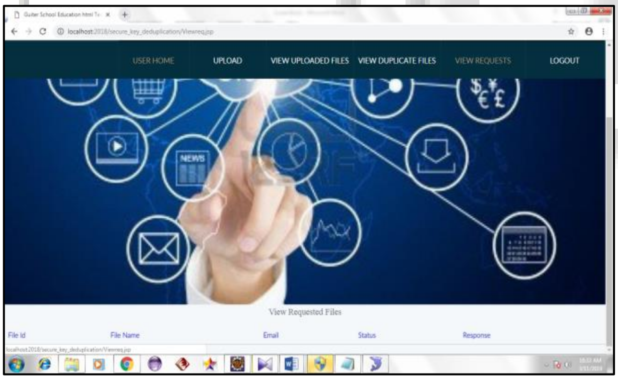
N. Owner View Duplicate Files



View Uploaded Files

File Id	File Name	Status	User email	Modify	Delete

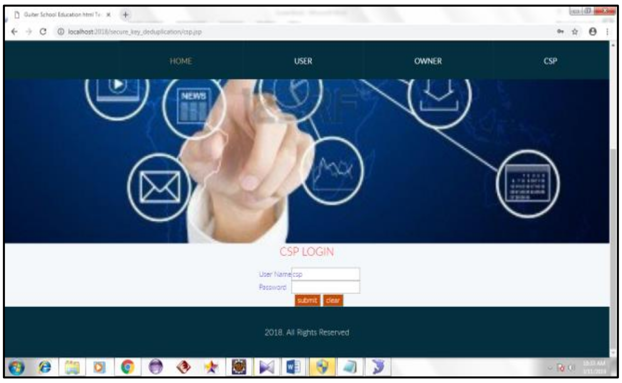
O. Owner View Requests


Guiter School Education html Te
localhost:2018/secure_key_deduplication/Viewreq.jsp
USER HOME
UPLOAD
VIEW UPLOADED FILES
VIEW DUPLICATE FILES
VIEW REQUESTS
LOGOUT
NEWS
localhost:2018/secure_key_deduplication/Viewreq.jsp

View Requested Files

File Id	File Name	Email	Status	Response

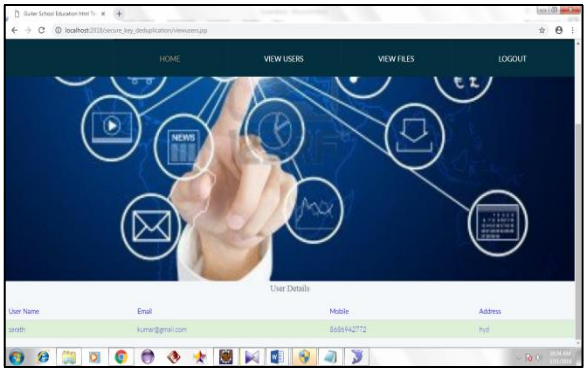
P. CSP Login


Guiter School Education html Te
localhost:2018/secure_key_deduplication/csp.jsp
HOME
USER
OWNER
CSP
NEWS
submit
clear
2018. All Rights Reserved

CSP LOGIN

User Name csp
Password

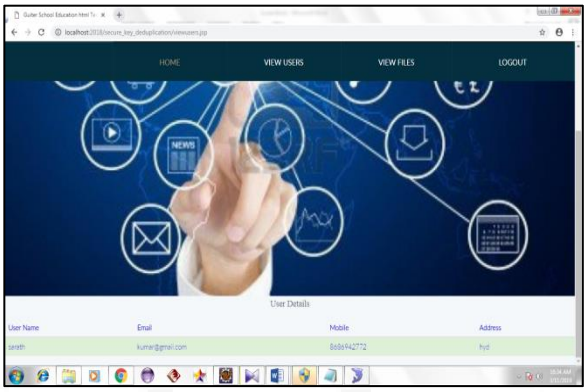
Q. CSP View Users


Guiter School Education html Te
localhost:2018/secure_key_deduplication/viewusers.jsp
HOME
VIEW USERS
VIEW FILES
LOGOUT
NEWS

User Details

User Name	Email	Mobile	Address
sarath	kumar@gmail.com	8686942772	hyd

R. CSP View Files


Guiter School Education html Te
localhost:2018/secure_key_deduplication/viewusers.jsp
HOME
VIEW USERS
VIEW FILES
LOGOUT
NEWS

User Details

User Name	Email	Mobile	Address
sarath	kumar@gmail.com	8686942772	hyd

VI. 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. 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.

REFERENCES

[1] D.A. Sarma, X. Dong, and A. Halevy, Bootstrapping pay-as-you-go data integration systems[C]. ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, Bc, Canada, June. DBLP, 2008:861-874.

[2] 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.

[3] S. Ghemawat, H. Gobioff, and S. Leung, The Google File System[M]. SOSP '03 Proceedings of the nineteenth ACM symposium on Oper- ating systems principles, 2003, 37(5): 29-43.

[4] D. Borthakur, HDFS architecture guide[J]. Hadoop Apache Project, 2008, 53.

[5] J. Li, X. Chen, M. Li, J. Li, P.P.C. Lee, and W. Lou, Secure Dedupli- cation with Efficient and Reliable Convergent Key Management[J]. IEEE transactions on parallel and distributed systems, 2014, 25(6): 1615-1625.

[6] G.R. Blakley and C.A. Meadows, Security of Ramp Schemes[C]. Crypto. 1984, 84: 242-268.

[7] A.D. Santis and B. Masucci, Multiple Ramp Schemes[J]. IEEE Trans- actions on Information Theory, 1999, 45(5): 1720-1728.