Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 6, Issue. 4, April 2017, pg.345 – 348

Honey Encryption Based Data Store in Cloud

D. Rekha[1], K. Selva Sheela[2]
[1]Computer science and Engineering, India
[2]Computer Science and Engineering, India

[1] rekha1105121@gmail.com; [2] k.selvasheela@gmail.com

***Abstract**— Cloud computing offers a new way of service provision by re-arranging various resources over the Internet. The most important and popular cloud service is data storage. In order to preserve the privacy of data holders, data are often stored in cloud in an encrypted form. However, encrypted data introduce new challenges for cloud data deduplication, which becomes crucial for big data storage and processing in cloud. Traditional deduplication schemes cannot work on encrypted data. Existing solutions of encrypted data deduplication suffer from security weakness. They cannot flexibly support data access control and revocation. It seems like no matter how much companies may try to up their defenses, there will always be some industrious young hacker who manages to evade every roadblock in his way. One team of researchers, though, think they may have finally found a way to turn their defense into an attack on the hackers themselves—by spewing fake data at them and sending them drowning. Currently, hackers will often use software that decrypts encrypted data by guessing hundreds of thousands of potential keys. So anytime an incorrect key is tried, the hackers are left with an incomprehensible mess that is distinctly* **not** *data and a clear indicator that the key or password was wrong. Still, if Honey Encryption works like its creators intend, this will definitely make hackers' jobs infinitely harder if not entirely impossible.*

I. Introduction

Cloud computing provides a big resource pool by linking network resources together. It has desirable properties such as scalability, elasticity, fault-tolerance, and pay-per-use. Thus, it has become a promising service platform. The most important and popular cloud service is data to data storage service. Cloud users upload personal or confidential data to the data center of a cloud service provider (CSP) and allow it to maintain these data. Since intrusions and attacks towards sensitive data at CSP are not avoidable, it is prudent to assume that CSP cannot be fully trusted by cloud users. Moreover, the loss of control over their own personal data, leads to high data security risks, especially data privacy leakages .due to the rapid development of data mining and others analysis technologies, the privacy issue becomes serious. Hence a good practice is to only outsource encrypted data to the cloud in order to ensure data security and user privacy. But the same or different users may upload duplicated data in encrypted form to CSP, especially for scenarios where data are shared among many users. Although cloud storage space in huge, data duplication greatly wastes network resource, consumes a lot of energy, and complicates data management.

Motivation

Cloud storage services are becoming very popular now a day. Cloud provides a better way of storage with efficient cost. One major problem with cloud is to manage huge amount of data. In order to manage data de-duplication technique is used. Although, de-duplication has many advantages but it has some security issues. This motivates us to propose a model which manage the security issues of de-duplication and provide authorized de-duplication in cloud.

In this paper, we propose a scheme based on data ownership challenge and Proxy Re-Encryption (PRE) to manage encrypted data storage with deduplication. We aim to solve the issue of deduplication in the situation where the data holder is not available or difficult to get involved. Meanwhile, the performance of data deduplication in our scheme is not influenced by the size of data, thus applicable for big data.

II. METHODOLOGY

Existing system: Deduplication to save space by storing only one copy of each file uploaded. However, if clients conventionally encrypt their data, storage savings by deduplication are totally lost. This is because the encrypted data are saved as different contents by applying different encryption keys. Existing industrial solutions fail in encrypted data deduplication. For example, is an efficient deduplication system, but it cannot handle encrypted data. The scheme mixes the randomly sampled portions of the original file with the dynamic coefficients to generate the unique proof in every challenge. The work focuses on ownership proof of the uploaded data during data deduplication. In this paper, we use Elliptic Curve Cryptography (ECC) to verify data ownership with the support of an authorized party.

Proposed system: Currently, hackers will often use software that decrypts encrypted data by guessing hundreds of thousands of potential keys. So anytime an incorrect key is tried, the hackers are left with an incomprehensible mess that is distinctly *not* data and a clear indicator that the key or password was wrong. So here going to use honey encryption to protect data from data owner and also the data accessing persons. And Access Point have separate key to providing to data to others. We propose a scheme to deduplicate encrypted data at CSP by applying PRE to issue keys to different authorized data holders based on data ownership challenge. It is applicable in scenarios where data holders are not available for deduplication control.

III.MODULE DESCRIPTION

System and Security Model:

The system contains three types of entities: 1) CSP that offers storage services and cannot be fully trusted since it is curious about the contents of stored data, but should perform honestly on data storage in order to gain commercial profits; 2) data holder (ui) that uploads and saves its data at CSP.

It seems like no matter how much companies may try to up their defenses, there will always be some industrious young hacker who manages to evade every roadblock in his way. One team of researchers, though, think they may have finally found a way to turn their defense into an attack on the hackers themselves—by spewing fake data at them and sending them drowning.

We propose a scheme to deduplicate encrypted data at CSP by applying PRE to issue keys to different authorized data holders based on data ownership challenge. It is applicable in scenarios where data holders are not available for deduplication control. A negative impact of bad reputation is the CSP will lose its users and finally make it lose profits. On the other hand, the CSP users (e.g., data holders) could lose their convenience and benefits of storing data in CSP due to bad reputation of cloud storage services. Thus, the collusion between CSP and its users is not profitable for both of them.

Verification of Data Ownership:

In order to check duplication, we first propose an ownership verification protocol based on a cryptoGPS identification scheme. AP to challenge data holder u_i to ensure that it is the real party that possesses data M. CSP just checks if duplication happens by verifying if the token $x_i = H(H(M) \times P)$ of data M has existed already. This design ensures that CSP cannot gain s_i and disclose H (M) to a malicious party.

Data Deduplication:

Data deduplication (often called "intelligent compression" or "single-instance storage") is a method of reducing storage needs by eliminating redundant data. Only one unique instance of the data is actually retained on storage media, such as disk or tape. Redundant data is replaced with a pointer to the unique data copy. For example, a typical email system might contain 100 instances of the same one megabyte (MB) file attachment. If the email platform is backed up or archived, all 100 instances are saved, requiring 100 MB storage space. With data deduplication, only one instance of the attachment is actually stored; each subsequent instance is just referenced back to the one saved copy. In this example, a 100 MB storage demand could be reduced to only one MB.

The benefits of compression and deduplication and explains how the two differ from each other. Data deduplication can generally operate at the file or block level. File deduplication eliminates duplicate files (as in the example above), but this is not a very efficient means of deduplication. Block deduplication looks within a file and saves unique iterations of each block. Each chunk of data is processed using a hash algorithm such as Honey Encryption Algorithm. This process generates a unique number for each piece which is then stored in an index. If a file is updated, only the changed data is saved. That is, if only a few bytes of a document or presentation are changed, only the changed blocks are saved; the changes don't constitute an entirely new file. This behavior makes block deduplication far more efficient. However, block deduplication takes more processing power and uses a much larger index to track the individual pieces.

Procedures of data deduplication

Step 1 – System setup.

Step 2 – Data token generation: User u1 generates data token of M, and sends to CSP.

Step 3 – Duplication check: CSP verifies and checks if the duplicated data is stored by finding

whether x1 exists. If the check is negative, it requests data upload. If the check is positive and the pre-stored data is from the same user, it informs the user about this situation. If the same data is from a different user.

Step 4 – Duplicated data upload and check: User u2 later on tries to save the same data M at CSP following the same procedure of Step 2 and 3.

Step 5 – Ownership challenge: AP challenges the data ownership of u2 i.e., the ownership challenge is successful, AP generates re-encryption

if it has not been generated and issued to CSP.

Step 6 – Deduplication: CSP re-encrypts. Then u2 can get DEK1 with its secret key sk2. u2 confirms the success of data deduplication to CSP that records corresponding deduplication information in the system after getting this notification.

Algorithm: Grant Access to Duplicated Data

```
Input: pkj, Policy(ui), Policy(AP)
- CSP requests AP to challenge ownership and grant access to
duplicated data for uj by providing pkj.
- After ensuring data ownership through challenge, AP checks
Policy(AP) and issues CSP rk(AP->ui)=RG if the check is positive.
- CSP transforms E(pkAP ;DEKi) into E(pkj;DEKi) if
Policy(ui) authorizes uj to share the same data M encrypted by
DEKi: R(rkAP->ui; E(pkAP ;DEK)) =E(pkj; DEKi).
Note: rkAP->ui calculation can be skipped if it has been executed
already and the value of (pkj; skj) and (pkAP ; skAP ) remain
unchanged.
- Data holder uj obtains DEKi by decrypting E(pkj; DEKi)
with skj: DEKi = D(skj;E(pkj); DEKi)), and then it can access
data M at CSP.
```

IV. CONCLUSIONS

The recent development of honey encryption offers many password based security schemes resilience to brute force offline attacks by yielding plausible plaintexts under decryption by invalid keys. Managing encrypted data with deduplication is important and significant in practice for achieving a successful cloud storage service, especially for big data storage. In this paper, we proposed a practical scheme to manage the encrypted big data in cloud with deduplication based on ownership challenge and PRE. We have presented our implementation of a honey encryption scheme and its application to a variety of use cases, ranging from generic alphabets to credit card numbers to text messaging. Specifically, we addressed the key challenge of generating plausible honey messages for each of these spaces by researching the probabilistic distribution of the message spaces and constructing good DTEs for each. Our scheme can flexibly support data update and sharing with deduplication even when the data holders are offline. Encrypted data can be securely accessed because only authorized data holders can obtain the symmetric keys used for data decryption. Extensive performance analysis and test showed that our scheme is secure and efficient under the described security model and very suitable for big data deduplication. The results of our computer simulations further showed the practicability of our scheme. Future work includes optimizing our design and implementation for practical deployment and studying verifiable computation to ensure that CSP behaves as expected in deduplication management.

REFERENCES

[1] M. Ali, et al., (2015) “SeDaSC: Secure data sharing in clouds,” IEEE Syst.J., vol. PP, no. 99, pp. 1–10.
[2] J. Li, Y. K. Li, X. F. Chen, P. P. C. Lee, and W. J. Lou, (2014) “A hybrid cloud approach for secure authorized deduplication,” IEEE Trans.Parallel Distrib. Syst., vol. 26, no. 5, pp. 1206–1216,.
[3] T. Y. Wu, J. S. Pan, and C. F. Lin, (2014) “Improving accessing efficiency of cloud storage using de-duplication and feedback schemes,” IEEE Syst. J., vol. 8, no. 1, pp. 208–218.
[4] P. Puzio, R. Molva, M. Onen, and S. Loureiro, (2013) “ClouDedup: Secure deduplication with encrypted data for cloud storage,’’ Proc. IEEE Int. Cof. Cloud Comput. Technol. Sci., pp. 363–370.
[5] J. W. Yuan and S. C. Yu, (2013) “Secure and constant cost public cloud storage auditing with deduplication,” n Proc. IEEE Int. Conf. Communic.Netw. Secur., pp. 145–153.
[6] C. Yang, J. Ren, and J. F. Ma, (2013) “Provable ownership of file in deduplication cloud storage,” in Proc. IEEE Global Commun. Conf., pp. 695–700.
[7] M. Bellare, S. Keelveedhi, and T. Ristenpart, (2013) “DupLESS: Server aided encryption for deduplicated storage,” in Proc. 22nd USENIX Conf. Secur., pp. 179–194.
[8] Z. C. Wen, J. M. Luo, H. J. Chen, J. X. Meng, X. Li, and J. Li, (2013) “A verifiable data deduplication scheme in cloud computing,” in Proc. Int. Conf. Intell. Netw. Collaborative Syst., pp. 85–90.
[9] M. Bellare, S. Keelveedhi, and T. Ristenpart, (2013) “Message-locked encryption and secure deduplication,” in Proc. Cryptology—EUROCRYPT, pp. 296–312.
[10] C. Y. Liu, X. J. Liu, and L. Wan, (2013) “Policy-based deduplication in secure cloud storage,” in Proc. Trustworthy Comput. Serv.,pp. 250–262.