ZKG INTERNATIONAL

ISSN: 2366-1313

Secure Data Deduplication with Encryption in Cyber Physical systems

[1]D. KAVITHA, [2]A. MANOJ KUMAR, [3]S. MALLESH, [4]V. SAHANA

[1]Assistant Professor, Dept.of IT, TKR College of Engineering & Technology, Meerpet, Hyderabad, davidikavitha2011@gmail.com

[2]BTech student, Dept.of IT, TKR College of Engineering & Technology, Meerpet, Hyderabad, manojarigachetta@gmail.com

[3]BTech student, Dept.of IT, TKR College of Engineering & Technology, Meerpet, Hyderabad, malleshsambu31@gmail.com

[4]BTech student, Dept.of IT, TKR College of Engineering & Technology, Meerpet, Hyderabad, srivallisahanaveleti@gmail.com

Abstract: *Cloud envisioned Cyber-Physical Systems (CCPS) is a practical technology that relies on the interaction among cyber elements like mobile users to transfer data in cloud computing. In CCPS, cloud storage applies data deduplication techniques aiming to save data storage and bandwidth for real-time services. In this infrastructure, data deduplication eliminates duplicate data to increase the performance of the CCPS application. However, it incurs security threats and privacy risks. For example, the encryption from independent users with different keys is not compatible with data deduplication. In this area, several types of research have been done. Nevertheless, they are suffering from a lack of security, high performance, and applicability. Motivated by this, we propose a message Lock Encryption with neVer-decrypt homomorphic EncRyption (LEVER) protocol between the uploading CCPS user and cloud storage to reconcile the encryption and data deduplication. Interestingly, LEVER is the first brute-force resilient encrypted deduplication with only cryptographic two-party interactions. We perform several numerical analyses of LEVER and confirm that it provides high performance and practicality compared to the literature.*

Keywords: *Cloud envisioned Cyber-Physical Systems, Cloud service providers, encryption, decryption, deduplication*.

I. INTRODUCTION

The amount of data to be stored by cloud storage systems increases extremely fast. It

ISSN: 2366-1313

is thus of utmost importance for Cloud Storage Providers (CSPs) to dramatically reduce the cost to store all the created data. A promising approach to achieve this objective is through data deduplication. Put simply, data deduplication keeps a single copy of repeated data. When a client wishes to store some piece of data, and if a copy of this data has already been saved in the storage system, then solely a reference to this existing copy is stored at the storage server. No duplicate is created [1]. There are diverse forms of data deduplication. It can be done by a client solely on the data he/she has previously stored in the system, a technique commonly called intra-user deduplication, or it can be achieved by taking into account the data previously stored by all the clients.

In this case it is designated as inter-user deduplication. Data deduplication also improves users experience by saving network bandwidth and reducing backup time when the clients perform the deduplication before uploading data to the storage server. This form of deduplication is termed as client-side deduplication, and when it is handled by the storage server it is called server-side deduplication. Due to its straightforward economic advantages, data deduplication is gaining popularity in both commercial and research storage systems[2].

Deduplication and confidentiality: the issues

Hereafter, we summarize in three categories the most common types of attacks described in the literature. Manipulation of data identifiers: A common technique to deduplicate data is by hashing the content of the data and using this unique hashed value as the identifier of the data.

Then the client sends this identifier to the storage server to determine whether such an identifier already exists in the system. An attacker can easily exploit this technique to perform various types of attacks. An example of attack based on data identifier manipulation is the CDN attack that turns a cloud storage system into a Content Delivery Network (CDN). Suppose that Bob wants to share a file F with Alice. Bob uploads F to a cloud storage system and sends F identifier to Alice. When Alice receives it, she tries to upload F to the same cloud storage system by sending F identifier [3].

The storage system will detect that this identifier already exists. Consequently, solely a reference meaning that Alice also owns file F will be stored in the system

ISSN: 2366-1313

which is actually wrong. At this point, when Alice wants to download F, she just needs to request it from the cloud storage system. Another example of attacks based on data identifier manipulation is the one called targeted collision in which, a malicious client uploads a piece of data (e.g, a file) that does not correspond to the claimed identifier. Suppose that Bob wants to cheat Alice. If no control is made by the cloud storage system, Bob can upload a file F1 with the identifier of a file F2. Then, if Alice wishes to upload F2 with its real identifier, the system will detect the existence of F2 identifier in the system and will not store F2. Rather, the system will store a reference meaning that Alice also owns file F1 which is the file corresponding to the identifier of F2 in the system. Later, when Alice will request the file corresponding to the identifier of F2, the system will send F1 to Alice.

Network traffic observation: Monitoring the network traffic on the client side gives an attacker the capability to determine whether an inter-user deduplication has been applied on a given piece of data. Such an attacker can exploit this knowledge to identify data items stored in a cloud storage system by other users or even learn the content of these data items.

For instance, to determine whether a file F is stored in a system, the attacker just tries to upload F and observes the network traffic from his/her device toward the storage server. If F is uploaded, it means that F does not exist in the storage system. The attacker is not even obliged to upload the complete file. He/she can prematurely abort the upload, so that the attack can be repeated later. On the other hand, if the file is not uploaded, the attacker can conclude that the file already exists in the system. This fact makes the storage system vulnerable to brute force attacks on file contents.

II. LITERATURE SURVEY

The survey on deduplication work with various algorithms tabulated them on the basis of algorithm, objective criteria, environment to which the works being performed. From the literature survey it is clear that, lot of work had been done already in deduplication but still it needs further development. (i.e.) Deduplication need to establish with high level security and minimum space wastage.

M. Dutch et al. [4] Data deduplication and other methods of reducing storage consumption play a vital role in affordably managing today's explosive growth of data. Optimizing the use of storage is part of a

ISSN: 2366-1313

broader strategy to provide an efficient information infrastructure that is responsive to dynamic business requirements. This paper will explore the significance of deduplication ratios related to specific capacity optimization techniques within the context of information lifecycle management.

D. T. Meyer and W. J. Bolosk [5] We found that whole-file deduplication achieves about three quarters of the space savings of the most aggres- sive block-level deduplication for storage of live file systems, and 87% of the savings for backup images. We also studied file fragmentation finding that it is not prevalent, and updated prior file system metadata studies, finding that the distribution of file sizes continues to skew toward very large unstructured files.

S. Halevi, D. Harnik et al. [6] In this work we identify attacks that exploit client-side deduplication, allowing an attacker to gain access to arbitrary-size files of other users based on a very small hash signatures of these files. More specifically, an attacker who knows the hash signature of a file can convince the storage service that it owns that file, hence the server lets the attacker download the entire file. (In parallel to our work, a subset of these attacks was recently introduced in the wild with respect to the Dropbox file synchronization service.)

D. Harnik, B. Pinkas et al. [7] Cloud storage refers to scalable and elastic storage capabilities delivered as a service using Internet technologies with elastic provisioning and usebased pricing that doesn't penalize users for changing their storage consumption without notice.

M. Mulazzani et al. [8] Within this paper we give an overview of existing file storage services and examine Dropbox, an advanced file storage solution, in depth. We analyze the Dropbox client software as well as its transmission protocol, show weaknesses and outline possible attack vectors against users. Based on our results we show that Dropbox is used to store copyright-protected files from a popular file sharing network. Furthermore, Dropbox can be exploited to hide files in the cloud with unlimited stor-age capacity

III. PROPOSED SYSTEM

Here, we present our message Lock Encryption with neVerdecrypt homomorphic EncRyption (LEVER). We divide the encrypted data deduplication into three phases; the user derives the chunk key for the chunk to be uploaded in the first phase, transforms the chunk to be uploaded into an encrypted form in the

ISSN: 2366-1313

second phase, and then runs the ordinary data deduplication protocol in the third phase. The design challenge of encrypted data deduplication is that a high min-entropy key can only encrypt the chunk. However, it is still difficult for different users with the same f to calculate the standard high min-entropy key presents the proposed cross-user cloud envision cyber-physical system (CCCPS) architecture.

As we can see, user 1 from CPS 1 and user 2 from CPS 2 upload the same file f into the CDD (see the continuous arrows in Fig. 1a). Meanwhile, an adversary aims to gain information about the file f and get access to the encrypted data to change its integrity in CDD. We require to impose our cloud server to preserve the data's security and privacy while satisfying the data deduplication technique in the system.

Our work takes place in the context of an Internet Service Provider (ISP) providing also the storage system2. That is to say, the ISP which is also the CSP has strong economic reasons to (i) save storage space and network bandwidth as it masters all the network and storage infrastructure, and (ii) provide a secure storage service to its consumers. We propose a deduplication solution to build a storage system that positively answers the following three questions:

1) Can we guarantee that the identifier used to trigger deduplication corresponds to the claimed data item?

2) Can we determine that a client actually owns the data item corresponding to the identifier issued to the storage system?

3) Can we make the inter-user deduplication transparent (i.e. unnoticeable even in the case of backup time or network traffic observation) to clients and still provide network bandwidth savings?

Specifically, our approach is a two-phase deduplication that leverages and combines both intra- and inter-user deduplication techniques by introducing deduplication proxies (DPs) between the clients and the storage server (SS). Communications from clients go through these DPs to reach the SS which allows splitting the deduplication process.

SYSTEM MODULE

Threat model

We suppose that the CSP is trusted to behave correctly to ensure the clients files availability and long-term durability, however it follows the honest but curious adversary model. Namely, it may take actions to disclose file contents. Regarding clients, we assume that some of them are malicious.

ISSN: 2366-1313

They may try to perform the types of attacks mentioned in Section I. We suppose that there is no coalition between the entities provided by the CSP (i.e. the SS and the DPs) and the clients. Finally, we consider that communication channels are reliable through the use of cryptographic protocols such as Secure Sockets Layer (SSL) or Transport Layer Security (TLS).

Data encryption

We make the hypothesis that entities have access to a hash function, denoted by hash in the following, that is pre-image resistant (i.e. given a hash value, it should be hard to find any file that hashes to that hash value), second pre-image resistant (i.e. given a file F1, it should be hard to find a different file F2 such that F1 and F2 hash to the same value), and collision resistant (i.e. it should be hard to find two different files F1 and F2 that hash to the same value). While the adoption of cloud storage services to outsource data is steadily growing, its benefits come with some inherent risks implied by the fact that full control on data is given to the CSPs]. Clients (e.g., individuals, scientists, enterprises, governmental agencies, etc) may want to restrict the control or access of the CSP to their sensitive data in order to preserve some confidentiality.

We assume that clients have the ability to encrypt and decrypt data with both asymmetric (encryptasym and decryptasym) and symmetric (encryptsym and decryptsym) cryptographic functions and manage their own private and public personal keys for the asymmetric encryption scheme. In our approach, clients encrypt their files using a convergent encryption scheme]. Specifically, to encrypt a file F with this scheme, the hashed value of F is used as an encryption key to encrypt F using a cryptographic symmetric function. Thus, the same file encrypted by different clients will result in equal ciphertexts. This allows the CSP to detect duplicates and to apply an inter-user deduplication on encrypted files without any knowledge of their plaintext contents.

User Interface Requirements

Administrative user interface

The ‘administrative user interface’ concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. These interfaces help the administrators with all the transactional states like Data insertion, Data deletion

ISSN: 2366-1313

and Date updating along with the extensive data search capabilities.

The operational or generic user interface

The 'operational or generic user interface' helps the end users of the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information in a customized manner as per the included Flexibilities.

IV. RESULTS

ARCHITECTURE DIAGRAM

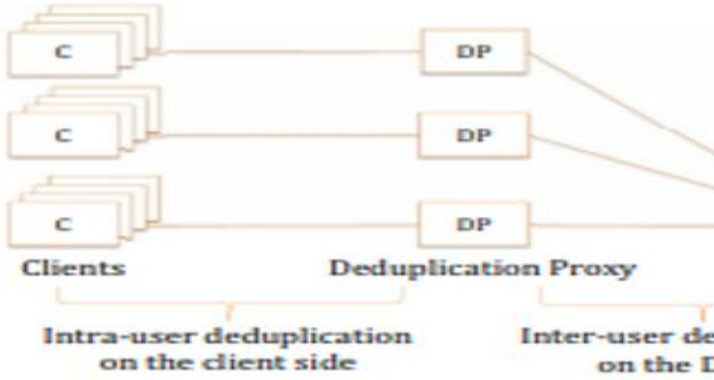

Fig.1 System architecture

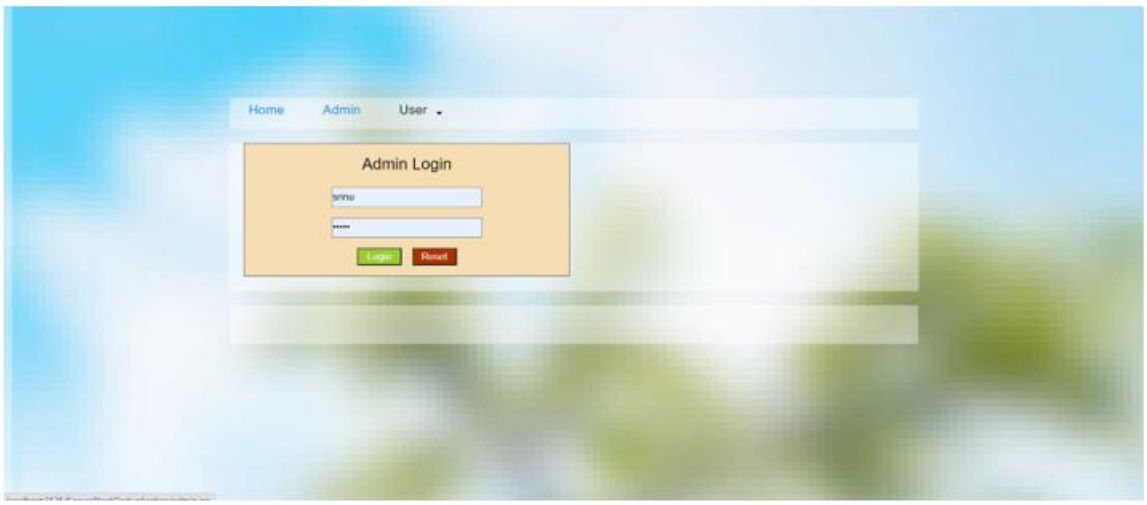

Fig.2 Admin page

ISSN: 2366-1313

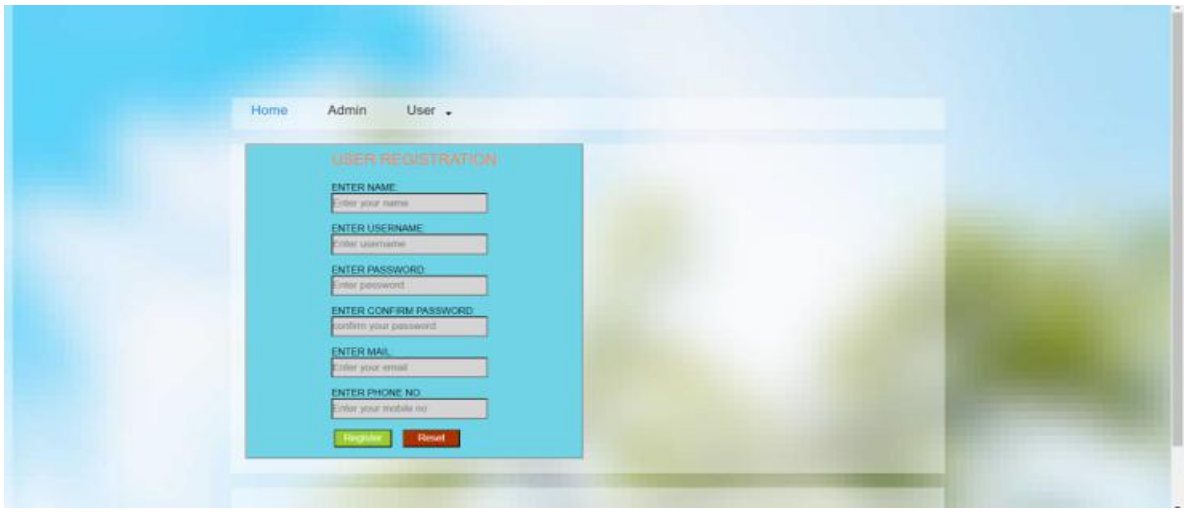

Fig.3 Registration page

User page

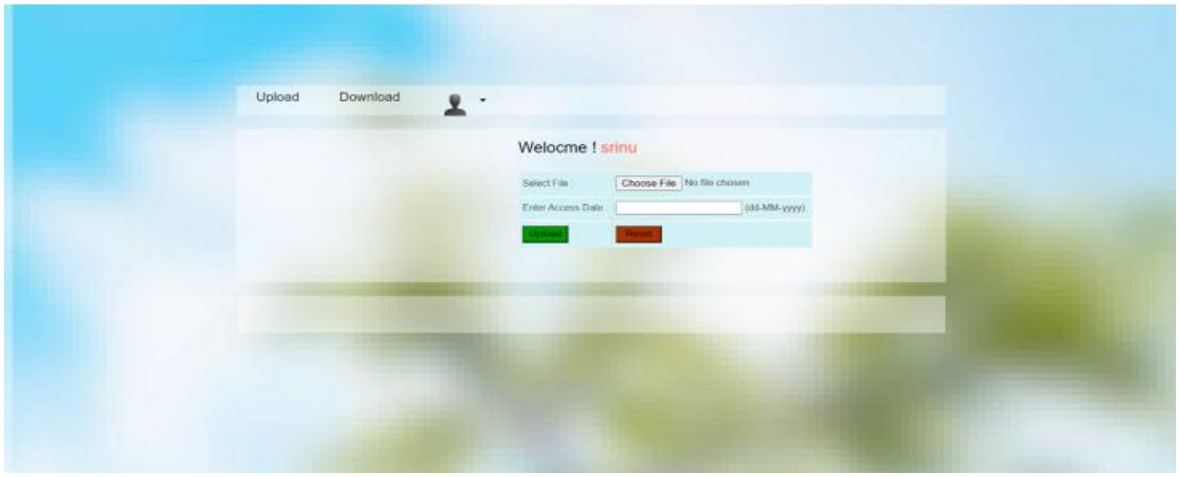

Fig.4 User page

ISSN: 2366-1313

Fig.5 Uploading page

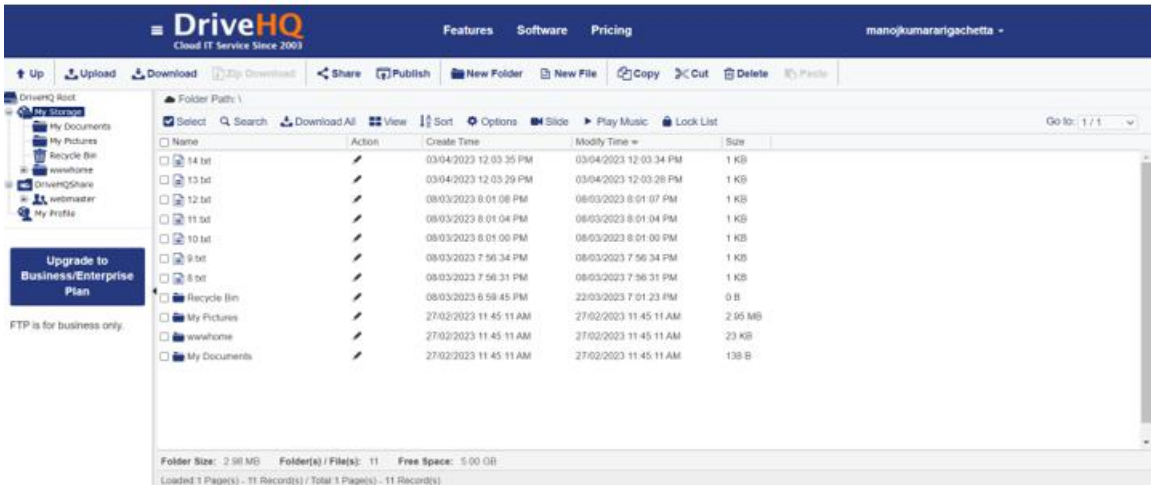

Fig.6 DriveHq Page

V. CONCLUSION

Lock Encryption with neVer-decrypt homomorphic EncRyption (LEVER) protocol between the uploading CCPS user and cloud storage to reconcile the encryption and data deduplication. Interestingly, LEVER is the first brute-force resilient encrypted deduplication with only cryptographic two-party interactions. We perform several numerical analyses of LEVER and confirm that it provides high performance and practicality compared to the literature. save storage space and network bandwidth as it masters all the network and storage infrastructure.

REFERENCES

[1] h. Song, D. Rawat, S. Jeschke, and C. Brecher, Cyber-Physical Systems: Foundations, Principles and Applications, 1st ed., ser. 1. Elsevier, 2016, academic Press.

[2] Y. Maleh, M. Shojafar, A. Darwish, and A. Haqiq, Cybersecurity and Privacy in Cyber Physical Systems, 1st ed., ser. 1. CRC Press, 5 2019.

[3] H. Song, G. Fink, and S. Jeschke, Security and Privacy in CyberPhysical Systems, 1st ed. Wiley Online Library, 2017.

[4] Y. Zhang, C. Xu, H. Li, K. Yang, J. Zhou, and X. Lin, "Healthdep: An efficient and secure deduplication scheme for cloud-assisted ehealth systems," IEEE Transactions on Industrial Informatics, vol. 14, no. 9, pp. 4101–4112, 2018.

[5] S. Garg, K. Kaur, N. Kumar, G. Kaddoum, A. Y. Zomaya, and R. Ranjan,

ISSN: 2366-1313

"A hybrid deep learning-based model for anomaly detection in cloud datacenter networks," IEEE Transactions on Network and Service Management, vol. 16, no. 3, pp. 924–935, 2019.

[6] S. S. Gill, S. Tuli, M. Xu, I. Singh, K. V. Singh, D. Lindsay, S. Tuli, D. Smirnova, M. Singh, U. Jain et al., "Transformative effects of iot, blockchain and artificial intelligence on cloud computing: Evolution, vision, trends and open challenges," Internet of Things, vol. 8, p. 100118, 2019.

[7] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in Annual international conference on the theory and applications of cryptographic techniques. Springer, 2013, pp. 296–312.

[8] N. Aviram, S. Schinzel, J. Somorovsky, N. Heninger, M. Dankel, J. Steube, L. Valenta, D. Adrian, J. A. Halderman, V. Dukhovni et al., "{DROWN}: Breaking {TLS} using sslv2," in 25th {USENIX} Security Symposium ({USENIX} Security 16), 2016, pp. 689–706.

[9] M. Bellare and S. Keelveedhi, "Interactive message-locked encryption and secure deduplication," in IACR International Workshop on Public Key Cryptography. Springer, 2015, pp. 516–538.

[10] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl, "A secure data deduplication scheme for cloud storage," in International conference on financial cryptography and data security. Springer, 2014, pp. 99–118.