ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 5, Issue 4, April 2018

Achieving Efficient and Privacy-Preserving Cross-Domain Big Data Deduplication in Cloud

[1] V.Neeraja, [2] N.Koteswara R, [3] T.Manjari, [4] P.Yamini, [5] V.Keerthana
[2] Associate professor Dept of CSE
[1][2][3][4] Narayana Engineering College, Gudur

Abstract: Secure data deduplication can significantly reduce the communication and storage overheads in cloud storage services, and has potential applications in our big data-driven society. Existing data deduplication schemes are generally designed to either resist brute-force attacks or ensure the efficiency and data availability, but not both conditions. We are also not aware of any existing scheme that achieves accountability, in the sense of reducing duplicate information disclosure (e.g., to determine whether plaintexts of two encrypted messages are identical). In this paper, we investigate a three-tier cross-domain architecture, and propose an efficient and privacy-preserving big data deduplication in cloud storage (hereafter referred to as EPCDD). EPCDD achieves both privacy-preserving and data availability, and resists brute-force attacks. In addition, we take accountability into consideration to offer better privacy assurances than existing schemes. We then demonstrate that EPCDD outperforms existing competing schemes, in terms of computation, communication and storage overheads. In addition, the time complexity of duplicate search in EPCDD is logarithmic.

INTRODUCTION

CLOUD storage usage is likely to increase in our big datadriven society. For example, IDC predicts that the amount of digital data will reach 44 ZB in 2020 [1]. Other studies have also suggested that about 75% of digital data are identical (or duplicate) [2], and data redundancy in backup and archival storage system is significantly more than 90% [3]. While cost of storage is relatively cheap and advances in cloud storage solutions allow us to store increasing amount of data, there are associated costs for the management, maintenance, processing and handling of such big data [4], [5]. It is, therefore, unsurprising that efforts have been made to reduce overheads due to data duplication. The technique of data deduplication is designed to identify and eliminate duplicate data, by storing only a single copy of redundant data. In other words, data deduplication technique can significantly reduce storage and bandwidth requirements [6]. However, since users and data owners may not fully trust cloud storage providers, data (particularly sensitive data) are likely to be encrypted prior to outsourcing. This complicates data deduplication efforts, as identical data encrypted by different users (or even the same user using different keys) will result in different cipher texts [7], [8]. Thus, how to efficiently perform data deduplication on encrypted data is a topic of ongoing research interest.
Next, we will introduce the system model, threat model and design goals in Section 2, before describing notations and bilinear groups of Composite order in Section 3. We then present the proposed EPCDD scheme in Section 4, analyze the scheme's privacy-preserving capability and security strength (data availability, accountability, and brute-force attack resilience) of the EPCDD scheme in Section 5, and demonstrate the efficiency of our EPCDD scheme in comparison to state-of-the-art schemes of [12], [14] in terms of computation, communication and storage overheads in Section 6. Related work is discussed in Section 7. We conclude this paper in Section 8.

MODELS AND DESIGN GOALS

In this section, we formalize the system model and threat model used in this paper, and identify our design goals.

System Model

The system model (see Fig.1) is a three-tier cross-domain big data deduplication system, which comprises a key distribution center (KDC), a cloud service provider (CSP), clients from different domains and the corresponding local managers, denoted as LMA and LMB. _ KDC: The trusted KDC is tasked with the distribution and management of private keys for the system.

CSP: The first tier is a CSP, which offers data storage services for clients. While the CSP is capable of supporting the storage needs of clients, it is financially vested to reduce the expensive big data management and maintenance overheads. Therefore, the CSP needs to perform inter-deduplication, which means that messages for deduplication are from different domains, to decrease the corresponding overhead.

ISSN (Online) 2394-2320

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

LMA (LMB): The second tier consists of domains (e.g., organizations such as companies or universities), which have cloud storage contracts with the CSP. Each domain maintains a local manager (e.g., LMA or LMB), which is responsible for intra-deduplication, and forwarding messages from clients in domain A (or B).

Clients: Every client is affiliated with a domain (e.g., employees in the company or students and faculty members in the university or university network, say University of Texas system). Clients upload and save their data with the CSP. In order to protect their data privacy and help the CSP to complete data deduplication over encrypted data, they encrypt the data and generate the corresponding tags. Finally, clients send message tuples containing encrypted data and the corresponding tags to the LMA or LMB (clients from domains A and B send message tuples to the LMA and LMB, respectively).

Threat Model
In our threat model, the CSP is considered honest but curious, which is the most common assumption in the literature (see [12], [14], [18]). Specifically, the CSP honestly follows the underlying scheme. However, it is curious about contents of stored data. Because the CSP adopts a pay-as-you-use model, it does not actively modify stored messages due to reputation, financial and legal implications (e.g. a civil litigation can result in significant reputation and financial losses to the provider). Hence, active attacks from the CSP are not considered in this paper. However, due to the significant amount of data stored in the cloud, it may know the plaintext space. Hence, according to the ciphertext and corresponding tags, the CSP (e.g. a malicious CSP employee) can carry out brute-force attacks. Finally, the CSP may obtain the plaintext corresponding to the special ciphertext for other illicit purposes (e.g. information reselling for financial gains).

LMA and LMB are also considered honest but curious. However, these entities have very limited computing and storage capabilities. Therefore, in practice, they do not have sufficient resources to carry out brute-force attacks. LMA or LMB may be curious about its affiliated clients' privacy, even though they may not actively seek to compromise the privacy of their clients. For example, if the domain is a company and LMA (or LMB) is the corresponding information manager. LMA (or LMB) is curious about the data uploaded by the staff. However, to protect the information asset, LMA or LMB does not actively attempt to compromise the privacy of clients, or collude with the CSP. Clients are considered honest. In theory, it is possible that they would collude with the CSP to obtain other clients' privacy. As mentioned in [14], in practice, such collusion may result in significant risks to the reputation of the CSP, as well as civil litigation or criminal investigations. In addition, if the CSP colludes with client A to compromise the privacy of client B, the CSP is also likely to collude with client B or other clients to compromise the privacy of other existing clients. This would have serious repercussions for the CSP if such collusion is reported or known. Thus, we assume that the CSP does not collude with its clients. Other than brute-force attacks, we do not consider other active attacks.

4. PROPOSED EPCDD SCHEME

In this section, we propose an efficient and privacy-preserving cross-domain deduplication scheme for big data storage (EPCDD).

4.1 Key Generation
KDC takes a security parameter _ as input, and outputs a 5-tuple (N; g;G;GT ; e) by running the composite bilinear parameter generator algorithm Gen(_). Then, it selects four random numbers s; t; a; b 2 ZN, where p j (as + bt), p - as and p - bt, and computes yA = gaq 2 G, yB = gbq 2 G. In addition, KDC chooses three cryptographic hash functions h1 : f0; 1g_ ! f0; 1gn, h2 : f0; 1g_ ! Z_ p and h3 : G ! f0; 1gn, where n is the bit length of symmetric key. Finally, KDC sends s and t to all members in domains A and B, respectively, and sends yA and yB to the CSP by secure channel. KDC publishes parameters pp = (N; g;G;GT ; e; e(g; g)s; e(g; g)t; h1; h2; h3).

4.2 Data Encryption and Tags Generation
For each client in domain A, after receiving the secret key s, the client encrypts the data mi and generates corresponding tags for data deduplication as follows.

4.2.1 Data Encryption
With secret key s and parameter e(g; g)t, the client computes the message-dependent symmetric key ski = h1(mike(g; g)st). Then, this client chooses a random number ri 2 ZN, computes the ciphertext Ci = Encski (rikmi), where the symmetric encryption algorithm is the cipher block chaining (CBC) mode, i.e., AESCBC.

4.2.2 Tags Generation
The client generates two tags for data mi as _ 1
i = gs_h2(mi),

_ 2 i = ski mod !, where ! is a random integer that not only ensures the security of ski but also meets the storage capacity of the CSP. For example, we can set j!j = 128 bits, thus, ski does not be disclosed by guessing attack if we set n = 256 bits (jskij=256), and it can represent up to 2128 data.
Similarly, clients in domain B execute same operations to generate ciphertexts and tags, e.g., for mj , encrypt it as Cj = Encskj (rjkmj), where skj = h1(mjke(g; g)st), and the corresponding tags are computed as _ 1
j = gt_h2(mj) 2 G,
_ 2
j = skj mod !.

PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed EPCDD scheme in terms of the computational, communication and storage overheads. Moreover, we give a comparison with the R-MLE2 (Dynamic) scheme [12] and Yan's scheme [14].

6.1 Computational Overheads

There are four entities in our EPCDD scheme, namely: clients, KDC, CSP and LMA (LMB). Under the aforementioned system model, KDC is responsible for generation of system parameters, which does not participate in the data deduplication. Thus, the computational overhead of the KDC can be ignored. We analyze the computational overhead of uploading one data in two cases: the data is duplicate and the data is new.
For data mi, the client first generates two tags _ 1 i and _ 2 to help the CSP to complete the deduplication. This operation requires one exponentiation in G, one multiplication in ZN, and one module in Z!. Since the multiplication in ZN and module in Z! are considered negligible compared to the exponentiation and pairing, the computational overhead of tag generation requires one exponentiation in G and one exponentiation in GT . If the duplication is found, the client does not need to do anything else. Otherwise, it needs to encrypt the data mi by symmetric encryption, i.e., AES-CBC. As shown in [14], the computational overhead of data encryption using symmetric encryption depends on the size of data, which is inevitable for protecting the data. Therefore, we can ignore the computational overhead of the encryption in
the comparison of these three schemes. In addition, regardless of whether duplicate data exist, the client needs to generate the message-dependent key ski for data availability, which costs one hash and one exponentiation in GT . It is worth noting that the computational overhead of hash depends on the size of data, but it is very fast, which can be ignored. Therefore, this computation only needs one exponentiation in GT . After receiving tags _ 1i and _ 2i , CSP compares _ 2i and current node ! _ 2 according to the DDT shown in Fig. 2. If _ 2i = current node ! _ 2, it needs to verify the Eq. (1), which requires two pairings and one multiplication in GT . As
shown in Algorithm 2, the CSP needs to verify the Eq. (1) if and only if _ 2
i = current node ! _ 2, which is independent of the search complexity. Hence, the computational complexity for
finding duplication is O(1). Moreover, for the intra-deduplication, LMA and LMB just compute the hash value h3(_ 1 i), and then check whether the same hash value exists (by comparing the value with the records in LMA or LMB). Obviously, the computational overhead for the LMA and LMB can be ignored.

CONCLUSION

Cloud storage adoption, particularly by organizations, is likely to remain a trend in the foreseeable future. This is, unsurprising, due to the digitization of our society. One associated research challenge is how to effectively reduce cloud storage costs due to data duplication.
In this paper, we proposed an efficient and privacy-preserving big data deduplication in cloud storage for a three-tier cross domain architecture. We then analyzed the security of our proposed scheme and demonstrated that it achieves improved privacy preserving, accountability and data availability, while resisting brute-force attacks. We also demonstrated that the proposed scheme outperforms existing state-of-the-art schemes, in terms of computation, communication and storage overheads. In addition, the time complexity of duplicate search in our scheme is an efficient logarithmic time.

REFERENCES

(1)IDC, "Executive summary: Data growth, business opportunities, and the it imperatives,," http:/ /www. emc. Com /leadership/ digital- universe / 2014iview /executive-summary. htm, 2014.

[2] J. Gantz and D. Reinsel, "The digital universe decadeare you ready," https: //hk. emc. com/ collateral/ analyst-reports/idc-digital-universe-are-you-ready.pdf.

[3] H. Biggar, "Experiencing data de-duplication: Improving efficiency and reducing capacity requirements," The Enterprise Strategy Group., 2007. [Online].Available: http:// journals. sagepub. com/doi /abs/10.1177/ 000944550704300309