International Journal of Research in Engineering, Science and Management
Volume-1, Issue-6, June 2018
www.ijresm.com

Enabling Secure and Effective Near Duplicate Detection Over Encrypted in Network Storage

Ratna[1], Punit Kumar[2]

[1,2]Student, Department of Studies in Computer Science & Engineering, UBDTCOE, Davanagere, India

***Abstract*—Near-duplicate detection (NDD) plays an essential role for effective resource utilization and possible traffic alleviation in many emerging network architectures, leveraging in-network storage for various content-centric services. As in network storage grows, data security has become one major concern. Though encryption is viable for in-network data protection, current techniques are still lacking for effectively locating encrypted near-duplicate data, making the benefits of NDD practically invalidated. Besides, adopting encrypted in network storage further complicates the user authorization when locating near-duplicate data from multiple content providers under different keys. In this paper, we propose a secure and effective NDD system over encrypted in-network storage supporting multiple content providers. Our design bridges locality sensitive hashing (LSH) with a newly developed cryptographic primitive, multi-key searchable encryption, which allows the user to send only one encrypted query to access near-duplicate data encrypted under different keys. It relieves the users from multiple rounds of interactions or sending multiple different queries respectively. As simply applying LSH does not ensure the detection quality, we then leverage Yao's garbled circuits to build a secure protocol to obtain highly accurate results, without user-side post-processing. We formally analyze the security strength. Experiments demonstrate our system achieves practical Performance with comparable accuracy to plaintext.**

Index Terms— near duplicate detection, network storage

I. Introduction

As indicated from EMC, the volume of digital data we create will grow exponentially in the next few years, reaching 44 zettabytes by 2020. To cope with such a huge amount of data, emerging network architectures, such as NDN and CDN, are proposed for content-centric applications, where the data are cached in globally distributed in-network servers. Accordingly, finding duplicate data, an indispensable technique in modern networked and storage systems, is naturally migrated to in-network servers from original data content providers to further alleviate network traffic and transmission latency. On the other hand, many types of data today, like images, videos, and web pages, are commonly seen to contain "essentially the same" content but differ in formats, encodings, or editing. As conventional deduplication via exact bit-streams matching can hardly identify them, techniques for effectively detecting such near-duplicate data can be absolutely necessary for high quality content-centric services, e.g., multimedia content delivery and sharing.

As more data are cached at in-network servers, they are also effectively becoming high-value target for both internal and external attacks. The content providers seek for strong protection of the copyrighted data against unauthorized content access or pirate, while the users do not want their accessed data being eavesdropped. While encrypting the data before deploying them in the network is a viable approach to address the security concerns, it would explicitly invalidate all the benefits of plaintext near-duplicate detection, preventing users from retrieving near-duplicates from nearby in-network servers. Besides, when utilizing encrypted in-network storage, another crucial requirement is to keep the user experience practically the same as in the plaintext domain. That is, when a user sends a query, the network should autonomously respond with the resulting near-duplicate data, ideally without multiple rounds of user interactions or multiple user queries sent subsequently. This, however, does not appear to be easy under an encrypted many-to-many scenario, where each in-network server may host encrypted data from multiple content providers under different keys, while processing the queries from different users.

II. Literature Survey

In the literature, NDD has been extensively studied in recent years. In general, NDD uses data-dependent features to characterize the data items, thereby being achieved via feature matching. Specifically, global feature based approaches use a compact fingerprint for a data item e.g., color histogram, while numerous local features e.g., SIFT, are adopted in the other ones. One recent work by Hua et al. proposed the concept of in-network de duplication for SDN, which eliminate duplicate data within the network by checking the data fingerprints in the SDN controller. In, they implemented an in-network near-duplicate detection scheme to support the image recovery in the context of disaster relief. Despite very useful, the above work operates in the plaintext domain. There are also a line of related designs (to just list a few) that study secure deduplication over encrypted data.

In Bellare et al. proposed the DupLESS that enables secure deduplication over encrypted documents with resilience to offline brute-force attacks. Later, Zheng et al. designed a secure layer-level deduplication system over encrypted scalable video coding videos for secure and efficient video delivery. Though considering privacy protection, these designs all focus on eliminating duplicates via exact fingerprint matching. Differently, our work targets a more general case, i.e., secure NDD. Our work is also related to multi-user searchable encryption that enables search over encrypted data for multi-user applications.

To extend symmetric key based searchable encryption in multi-user scenarios, Curtmola et al. proposed to use broadcast encryption to share a randomness among all legal users for the authentication of search tokens. Jarecki et. al.proposed to require legal users to obtain the search tokens from the data owner whenever they want to search. Though supporting multiple users search, these schemes only support one data owner. Namely, users need multiple search tokens for a given query when there are multiple data owners. On the other hand, asymmetric key based approaches also examine the multi-user setting. Boneh et al. scheme that supports multiple data owners.

As long as the user has the private key, she can search and decrypt the data. Likewise, if she wants to search multiple repositories, multiple tokens are required as well.

III. Proposed System

We propose a secure and effective system that enables in-network servers to locate encrypted near-duplicates for authorized users across multiple content providers. We design an authorized NDD algorithm over encrypted data running on a single server, and extend it by proposing a protocol across distributed in-network servers. We also design and implement a secure two-party computation protocol based on Yao's garbled circuits to obtain accurate detection results. We formally analyse the security strength, and implement all components in a cloud-based prototype system. Extensive experiments on real-world dataset show that our system achieves.

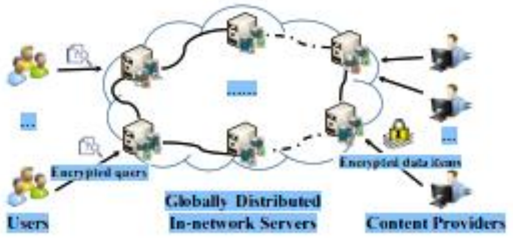

Fig. 1. Architecture diagram

Our system consists of three major entities: the content providers (abbr. CPs), the in-network servers (abbr. ISs), and the users, as shown in Fig. 1. Considering a practical scenario, CPs intend to outsource the data to globally distributed ISs for high quality of content hosting and delivery services. Since ISs are usually deployed in untrusted network environments by third-party service providers, e.g., CDN and Cloud, CPs will encrypt the data against data leakage and unauthorized access. For bandwidth efficiency, they still demand ISs to conduct secure and accurate NDD over encrypted in-network storage.

On the other hand, a certain IS may host data items from different CPs, which are encrypted with their own keys. As a result, applying existing efficient searchable encryption schemes [12], [19] for secure NDD will force the user to generate multiple encrypted queries, if she is authorized by different CPs. In addition, if the near-duplicate data are not found "nearby", all the queries would be propagated or resent to the next IS. From the user's perspective, the above inefficiency should also be addressed. Our design aims to allow the user to send one encrypted query only, comparable to the situation in the plaintext domain, while secure NDD can still be conducted over encrypted data items from different CPs. The service flow of our system consists of the three phases:

1. Preparation phase
2. Detection phase
3. Evaluation phase

1) Preparation phase: In this phase, two procedures are executed. (1) The CPs encrypt the data items and prepare the cipher text metadata fcg, where c is derived from each data item, and used for later secure NDD. We use "fg" to represent a collection. The network service provider assigns the encrypted data items with fcg to ISs that are close to the users. (2) The CPs authorize their users so that fcg is accessible to legal users. Explicitly, each user generates her own key, and each CP generates an authorization digest for every user from the user key. As long as an IS hosts the encrypted data from a certain CP, the corresponding's for users of that CP are transmitted to the IS. Here, we emphasize that the above authorization is fully distributed without relying on any trusted authority for key distribution.

2) Detection phase: The user generates an encrypted query q from the data of interest via her own key, and sends it to "nearby" IS. The IS will first check whether the user has access privileges, i.e., authorization digests _s, to access the data from different CPs. If "a" exists, tq will be transformed into the form that can be tested with the cipher text metadata fcg from the corresponding CP. When a match is found, the encrypted data item will be considered as a near-duplicate candidate. Otherwise, tq will be forwarded to the next IS until locating sufficient near-duplicate candidates. In this phase, the user just needs to send one tq only, which will be forwarded and transformed by ISs without any interaction with the user.

3) Evaluation phase: To eliminate the false positives from the initial detection results, a secure two-party computation protocol is initiated between the corresponding IS and a garbled-circuit generator. Here, the generator can be a server from another service provider, which does not collude with the IS. Specifically, the generator prepares a garbled circuit for the IS, which securely evaluates whether the distances between the encrypted fingerprints of candidates and the query data item are within a pre-defined threshold. The candidates that satisfy the evaluation will be directly sent back to the user as the final near-duplicate results.

IV. Conclusion

In this paper, we propose a secure and effective NDD system over encrypted in-network storage supporting multiple content providers. In order to relieve the users from multiple rounds of interactions or sending multiple different queries, our design bridges LSH with a newly developed cryptographic primitive MKSE, which allows the user to send only one encrypted query to access near-duplicate data encrypted under different keys. Furthermore, to ensure the quality of secure NDD, we leverage a secure two-party computation protocol based on Yao's garbled circuits to obtain highly accurate results without user-side post-processing. We formally analyze the security strength. Extensive experiments on real-world dataset demonstrate that our system achieves practical performance with comparable accuracy to plaintext. As future work, we will conduct thorough security analysis of the system. We will investigate the possible extension in other realistic scenario, e.g., querying a targeted in-network server, and even performance speedup via batch processing.

References

[1] IDC, "Executive Summary: Data Growth, Business Opportunities, and IT Imperatives," Online at http://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm.

[2] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos,L. Wang, B. Zhang et al., "Named data networking," ACM SIGCOMM Computer Comm. Review, vol. 44, no. 3, pp. 66–73, 2014.

[3] Akamai, "Akamai," Onlineathttps://www.akamai.com/, 2015.

[4] Y. Hua, X. Liu, and D. Feng, "Smart in-network deduplication for storage-aware sdn," in Proc. of ACM SIGCOMM, 2013.

[5] Y. Hua, W. He, X. Liu, and D. Feng, "Smarteye: Real-time and efficient cloud image sharing for disaster environments," in Proc. of IEEE INFOCOM, 2015.

[6] Y. Ke, R. Sukthankar, and L. Huston, "Efficient near-duplicate detection and sub-image retrieval," in Proc. of ACM MM, 2004.

[7] B. Thomee, M. J. Huiskes, E. Bakker, and M. S. Lew, "Large scale image copy detection evaluation," in Proc. of ACM international conference on Multimedia information retrieval, 2008.

[8] X. Wu, A. G. Hauptmann, and C.-W. Ngo, "Practical elimination of near-duplicates from web video search," in Proc. of ACM MM, 2007.

[9] X. Yuan, X. Wang, C. Wang, A. Squicciarini, and K. Ren, "Enabling privacy-preserving image-centric social discovery," in Proc. of IEEE ICDCS, 2014.

[10] W. Sun, S. Yu, W. Lou, Y. T. Hou, and H. Li, "Protecting your right: Attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud," in Proc. of IEEE INFOCOM, 2014.

[11] A. Gionis, P. Indyk, R. Motwani et al., "Similarity search in high dimensions via hashing," in Proc. of VLDB, 1999.

[12] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in Proc. of ACM CCS, 2006.

[13] M. Kuzu, M. Islam and M. Kantarcioglu, "Efficient similarity search over encrypted data," in Proc. of IEEE ICDE, 2012.

[14] H. Cui, X. Yuan, and C. Wang, "Harnessing encrypted data in cloud for secure and efficient image sharing from mobile devices," in Proc. Of IEEE INFOCOM, 2015.

[15] M. Li, S. Yu, N. Cao, and W. Lou, "Authorized private keyword search over encrypted data in cloud computing," in Proc. of IEEE ICDCS, 2011.

[16] R. A. Popa and N. Zeldovich, "Multi-key searchable encryption," IACR Cryptology ePrint Archive, 2013.

[17] C. Liu, X. S. Wang, K. Nayak, Y. Huang, and E. Shi, "Oblivm: A programming framework for secure computation," in Proc. of IEEE Security and Privacy (S&P), 2015.

[18] V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, and N. Taft, "Privacy-preserving ridge regression on hundreds of millions of records," in Proc. of IEEE Security and Privacy (S&P), 2013.

[19] S. Jarecki, C. Jutla, H. Krawczyk, M. Rosu, and M. Steiner, "Outsourced symmetric private information retrieval," in Proc. of ACM CCS, 2013.

[20] R. A. Popa, E. Stark, J. Helfer, S. Valdez, N. Zeldovich, M. F. Kaashoek, and H. Balakrishnan, "Building web applications on top of encrypted data using mylar," in Proc. of USENIX NSDI, 2014.

[21] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in Proc. of EUROCRYPT, 1999.

[22] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Dupless: Server-aided encryption for deduplicated storage," in Proc. of USENIX Security, 2013.

[23] Y. Zheng, X. Yuan, X. Wang, J. Jiang, C. Wang, and X. Gui, "Enabling encrypted cloud media center with secure deduplication," in Proc. Of ASIACCS, 2015.

[24] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Proc. of EUROCRYPT, 2004.

[25] F. Bao, R. H. Deng, X. Ding, and Y. Yang, "Private query on encrypted data in multi-user settings," in Proc. of ISPEC, 2008.

[26] C. Dong, G. Russello, and N. Dulay, "Shared and searchable encrypted data for untrusted servers," Journal of Computer Security, vol. 19, no. 3, pp. 367–397, 2011.