International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2016 IJSRCSEIT | Volume 1 | Issue 3 | ISSN : 2456-3307

MPPSE : Multi-Source Privacy-Preserving Symmetric Encryption scheme over Personal Health Record

[1]Banoth Seetha Ramulu, [2]H. Balaji

[1]Associate Professor, Department of CSE,Vardhaman College of Engineering, Shamshabad, Hyderabad, TS, India

[2]Associate Professor, Department of CSE, Sreenidhi Institute of Science and Technology, Ghatkesar, Hyderabad, TS, India

ABSTRACT

Cloud-Centered Personal Health Record systems (CC-PHR) have great prospective in enabling the managing of individual health records. Security and privacy concerns are among the main obstacles for the wide adoption of CC-PHR systems. In this paper, we consider a multi-source CC-PHR system in which multiple data providers such as hospitals and physicians are authorized by individual data owners to upload their personal health data to an untrusted public cloud. The health data are submitted in an encrypted form to ensure data security, and each data provider also submits encrypted data indexes to enable queries over the encrypted data. We propose a unique Multi-Source Privacy-Preserving Symmetric Encryption (MPPSE) scheme whereby the cloud can merge the encrypted data indexes from multiple data providers without knowing the index content. MPPSE enables efficient and privacy-preserving query processing in that a data user can submit a single data query the cloud can process over the encrypted data from all related data providers without knowing the query content. We also propose an enhanced scheme, MPPSE+, to more efficiently support the data queries by hierarchical data providers. Extensive analysis and experiments over real datasets demonstrate the efficacy and efficiency of MPPSE and MPPSE+.

Keywords : Authorization Query, Cloud Computing, Personal Health Record, Privacy-Preserving Query

I. INTRODUCTION

Cloud-Centered Personal Health Record systems (CC-PHR) such as Microsoft HealthVault1 and ZebraHealth2 are rising. A typical CC-PHRsystem consists of three entities: data owners, data providers and a cloud server. In CBPHR system, data owners and data providers are defined as patients themselves and hospitals, respectively. Data owners can directly authorize data providers to upload their PHRs to the cloud. The CC-PHRsystem allows data owners to access their PHRs anytime and anywhere, is better prepared for medical appointments and unexpected emergencies, maintain a more complete picture about personal health, and even achieve fitness goals. Data providers can explore the CB1http://www.healthvault.com 2https://www.zebrahealth.com PHR system to provide better medical services by sharing, collaborating, and engaging with the patients in new ways. Privacy concerns are among the main obstacles for the wide adoption of CC-PHRsystems. In particular, many people have deep concerns that there can be unauthorized access to their sensitive PHRs. For example, the cloud may have business

interest in analyzing the PHRs, and it may also have malicious employees or even be hacked.

A natural way to alleviate the privacy concerns is to let data owners and providers upload encrypted PHRs to the cloud which does not possess the decryption keys [1]–[5], [8]. Since PHRs can be in huge volume, it is very inefficient for data owners or providers to retrieve all the encrypted PHRs from the cloud when only a small portion of them are needed. To enable efficient queries over encrypted PHRs, the B+-tree technique [2]–[5], [9] is proposed to build an index for each patient's PHRs. The data index allows the cloud server to quickly find all the PHRs matching a particular data query. To further resolve the privacy concerns about data indexes and queries, searchable encryption schemes [10]–[20] are proposed to encrypt data indexes and queries as well. These schemes allow the cloud server to perform efficient queries over encrypted PHRs directly based on the encrypted indexes and queries while blind to the index and query content. Traditional searchable encryption schemes [10]–[20] are designed for generic cloud platforms and not optimized for CC-PHRsystems. In particular, the PHRs of different data providers for the same data owner may be highly correlated and associated with the same attributes (e.g., symptoms). If a traditional search encryption scheme is used, each data provider needs to independently generate the encrypted data index for submission to the cloud server. Therefore, the data owner needs to manage the keys with different data providers and also submit a dedicated data query for each data provider even if query conditions are exactly the same.

A plausible solution to this issue is to let all the data providers use a common key assigned by the data owner to encrypt the data indexes associated with him.3 This method, however, is vulnerable to the compromise of a single data provider. In this paper, we propose a very efficient PHR system with strong privacy guarantees. In our system, each data owner authorizes multiple data providers to submit encrypted health records and data indexes to the cloud server. Our system differs from prior work in two desirable features. First, each data provider of the same data owner uses a unique symmetric key for encrypting data indexes, thus resisting single point of compromise. Second, each data owner needs not manage the keys with individual health providers and can submit a single encrypted query to the cloud server for searching over the encrypted health data from all his data providers. The second feature enables very efficient query processing.

II. SYSTEM AND THREAT MODELS

A. System Model and Workflow

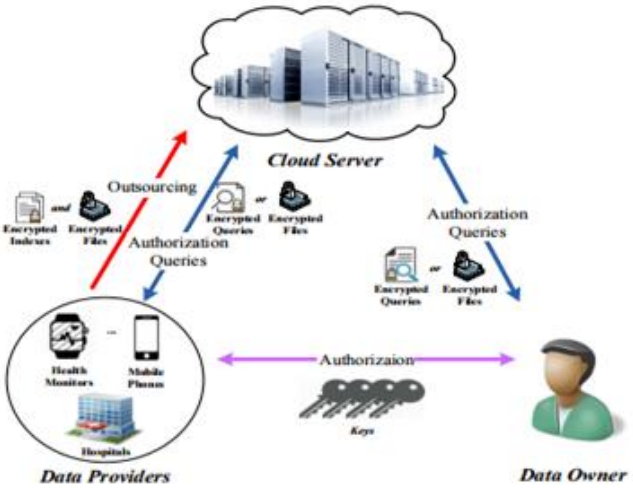

Figure 1. System Model

We consider a generic CC-PHRsystem shown in Fig. 1. There are three kinds of entities: the cloud server, data owners, and data providers. A data owner refers to a patient who owns the PHRs. In contrast, a data provider can refer to a patient himself, any of his health providers such as a physician or hospital, and even his personal health monitoring device. The cloud server stores and provides anytime, anywhere access to the PHRs submitted by the data providers of each data owner. Each data owner has strong privacy concerns for his PHRs. His data providers thus must encrypt the PHRs before outsourcing them to the cloud server. To ensure efficient search for the encrypted PHRs, each data provider additionally

uploads a data index to the cloud server. The data owner or any of his authorized data providers can submit data queries to the cloud server. Both data indexes and queries should be encrypted as well to prevent information disclosure. The cloud server explores the data indexes to locate the PHRs satisfying each query without the capability or need to decrypt the PHRs, data indexes, or data queries. Finally, the cloud server returns the corresponding encrypted PHRs to the requesting data user who can decrypt them with the right decryption key.

B. Threat Model

We assume a conventional threat model as follows. The cloud server is honest-but-curious by faithfully running the system but having strong interest in the content of the PHRs, data indexes, and queries. We also assume that data providers are untrusted and may try to acquire the PHRs generated by other providers. Besides, we assume that the communications within our system are secured using traditional mechanisms such as TLS (Transport Layer Security) [21]. Finally, we assume that the cloud server does not collude with data providers to compromise data owners' privacy. The privacy of data owners can be classified into PHR privacy, index privacy, and query privacy. Since our system stores encrypted PHRs at the cloud server which has no decryption keys, PHR privacy is easily achieved as long as the underlying encryption primitive is unbreakable. We thus focus on index privacy and query privacy hereafter. Index privacy is considered compromised if the content of any encrypted index is known to the cloud server or any data provider other than the source data provider. In contrast, query privacy is said to be breached in either scenario below. First, the content of any encrypted query is disclosed to anyone other than the data user (i.e., the data owner or any of his authorized data providers). Second, a data provider generates a valid query without obtaining the authorization of the data owner.

III. IMPROVED SCHEME

In previous model, data providers (e.g., the hospitals and personal health monitors) can only access the patient's data issued by them. However, in reality, data providers may access health data generated by other data providers. For instance, research-oriented hospitals utilize patients' data to prevent the incidence of common diseases; the doctors may access the data recorded by personal health devices. Thus, considering a hierarchical system model will make our model more practical. In our hierarchical model, data providers have various privileges according to the actual requirements. For example, personal health devices are mostly with the lowest privileges, because of being accessed in common; research oriented hospitals are with higher privileges than community hospitals, since the formers need amount of patients' data to do research, while community hospitals only access specific patients' data. With the basic solution MPPSE, we can implement hierarchical structure for data providers, i.e., issuing all keys of data providers with lower privileges to those with higher privileges. However, the solution is inefficient, such as the overhead of the management of keys increases linearly with the number of data providers with lower privileges, and data providers need to generate amount of queries (the number is positively related to providers' number). To thwart these inefficient problems, we proposed an enhanced scheme (named MPPSE+) based on our previous scheme MPPSE[27].

IV. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of the MEIM mechanism by three schemes: OPSE, MPPSE and MPPSE+. We both implement schemes in C++, and MPPSE+ is with the Pairing-Based Cryptography (PBC) Library. Note that the type A elliptic curve parameter is adopted, where the group order is 160-bits, providing 80-bit security strength equivalently. For experiment dataset, since no real PHR datasets are publicly available for academic purposes, we

apply MEIM to Nursery Dataset obtained from the UCI Machine Learning Repository [22]. The dataset is used in the previous research on searchable encryption [3], [12]. Briefly, the dataset comprises 12,960 instances, which contain eight categories reaching up to five values each. These experiments are carried out on an IBM workstation running Red-Hat Linux with a Intel(R) Xeon(R) CPU E5606 (with four cores) @2.13GHz with 12GB of random access memory. The experimental results show that these algorithms on MEIM perform well.

In MPPSE, categories and values are deemed to be attributes and attribute values, respectively. Each value in Nursery Dataset will be coded into integers by Unicode. However, in MPPSE+, we follow the experimental setup to set privileges. Namely, each instance is randomly defined as a hth privilege, where the front h categories are utilized to generate the vectors of the secret/public keys for the corresponding h privilege, and the rest categories are viewed as the random vectors. Here, the category values are converted into elements in Fϕ using SHA-1 hash algorithm. To evaluate the efficiency of our schemes for various instances' number, we divide the data set into ten subsets, and each contains 1296 instances. In the evaluation of the encryption and query efficiency, we utilize multiple subsets, from one to ten, to test the encryption and query time. Note that our dataset are organized by MDBT, and stored in memory. The following experimental results are based on this premise[26].

A. INDEX GENERATION

To evaluate the performance of index generation, we utilize n subsets to generate n indexes with MDBT. Each MDBT contains eight layers corresponding to eight attributes, respectively. According to the performance analysis of index generation, we obtains two conclusions: (1) the time complexity of MPPSE is equal to that of MPPSE+; (2) the time complexity of generating a index is $O(\psi \cdot \tau \cdot \gamma)$. Fig. 6a and 6b present the average running time and memory consumption for generating n indexes ($1 \leq n \leq 10$). In our experiment, we set ψ as 8, thus the generation time of indexes increases with the multiple of τ and γ. As shown in Fig. 2a, when the index numbers pick up {3, 7, 9}, the time overhead are {8.845ms, 27.849ms, 39.080ms}. Meanwhile, the points of MPPSE and MPPSE+ are overlap, which satisfies the conclusion (2). For memory usage of indexes, Fig. 2b shows that it is linear growth with an increasing number of n. When n is 2, the memory usage is 20.25KB.

B. INDEX ENCRYPTION

To evaluate the performance of index encryption, we take the above indexes as input, and runs MPPSE and MPPSE+ schemes to output the encryption time. In MPPSE, we utilize the AES scheme as a symmetric encryption, and HMACKi (·) to generate Mi,d. As shown in Figure 2d, the encryption overhead for indexes increases with the indexes number n. When n=3, the encryption time is about 1289.894ms.

In MPPSE+, we use our proposed method HPBPE to generate Ch,d. Figure 3a shows the time overheads for setup, key generation and delegation in MPPSE+. In setup, the overhead includes O(z 2 0) = O(z 2) exponentiations each (where z0 is equivalent to z+3). When z is 31, the setup time is about 1.56s. To evaluate the performance of key generation and delegation, we choose h ($1 \leq h \leq 8$) identification from the identification universe in each identification to form a query. That is, the vector ˜v does not have element $0 \in F\phi$. According to Figure 3a, the direct key generation consumes relatively long time, while the delegation consumes less time. The reasons are that the former is processed by the central trusted authority because it is usually an one-time operation; while the latter is experienced by Level-2 local trusted authority and users under a Level-1 local trusted authority. Notice that the capability generation/delegation times both scale as O(z 2 0). Fig. 6e shows the encryption consumption with MPPSE+, and we conclude that the encryption

overhead in MPPSE is lower than that in MPPSE+ through comparing Figure 6d and 6e[25].

For a attribute value, MPPSE+ takes 42.5ms to encrypt it. For memory cost in MPPSE, the label for a attribute value consists of two parts: Mi,d and Pi,d. Each prefix in Mi,d is with 128-bit, and the Pi,d is a integer value. While in MPPSE+, the memory overhead of Ch,d need 65(z0+1)B. For a attribute value, when z picks up 46 , the size is equal to merely 3.2KB. Figure 6c shows the comparison of the memory overhead between MPPSE and MPPSE+.

C. ENCRYPTED INDEXES TRANSFORMING

To evaluate the overhead of the merging for multiple encrypted indexes, the cloud server merges the aforementioned encrypted indexes due to the comparison of the suffixes. Fig. 3b concludes that the merging time grows linearly with the elements number in the root which is reasonable because ciphertexts for the same plaintext on different indexes are unequal. When n is {5, 10}, the merging time is {4us, 15us}, respectively. According to the aforementioned analysis, ciphertexts for one plaintext are distinct, thus the merging cannot lead to a changeable in memory overhead. While for the segmenting, the consumption is a positive correlation with the total numbers of the nodes[24].

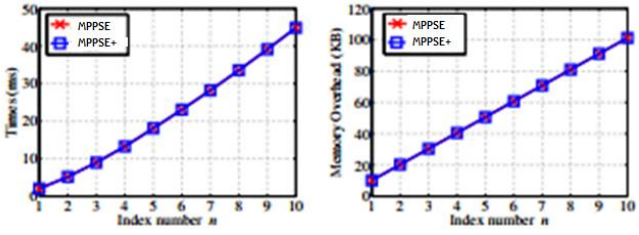

Figure 2(a) Index generation time
2(b) Index Memory overhead

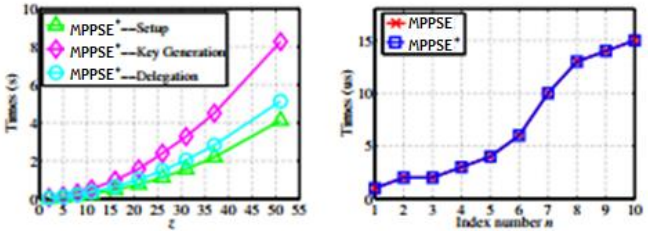

Figure 3(a) Time over head in MPPSE+
3(b) Index merging time

V. CONCLUSION AND DISCUSSION

In this paper, we explore the problem of privacy-preserving query for multi-source in the cloud-based PHR environment. Different from prior works, our proposed MEIM mechanism enables authenticated data owner to achieve secure, convenient, and efficient query over multiple data providers' data. To implement the efficient query, we introduce MDBT as the data structure. To reduce the overhead of query generation of data owner, and allow the cloud server to securely query, we propose a novel multiple order-preserving symmetric encryption (MPPSE) scheme. To make our model more practical, we propose an enhanced multiple order-preserving symmetric encryption (MPPSE+) scheme to satisfy the hierarchical authenticated query. Moreover, we leverage rigorous security proof to prove that our schemes are security. Finally, we demonstrate that the MEIM mechanism is computationally efficient by implementing our schemes and running in a real dataset.

VI. REFERENCES

[1] C. Wang, B. Zhang, K. Ren, J. Roveda, C. Chen, Z. Xu, "A privacy-aware cloud-assisted healthcare monitoring system via compressive sensing," in INFOCOM'14, Toronto, Canada, 2014.

[2] J. Sun, X. Zhu, C. Zhang, Y. Fang, "HCPP: Cryptography based secure ehr system for patient privacy and emergency healthcare," in ICDCS'11, Minneapolis, Minnesota, 2011.

[3] M. Li, S. Yu, N. Cao, W. Lou, "Authorized private keyword search over encrypted data in cloud computing," in ICDCS'11, Minneapolis, Minnesota, 2011.

[4] J. Benaloh, M. Chase, E. Horvitz, K. Lauter, "Patient controlled encryption: ensuring privacy of electronic medical records," in: ACM workshop on CCS'09, New York, NY, 2009.

[5] M. Li, S. Yu, Y. Zheng, K. Ren, W. Lou, "Scalable and secure sharing of personal health

records in cloud computing using attribute-based encryption," IEEE T Parall Distr., vol. 24, no. 1, pp. 131 - 143, 2013.

[6] M. Li, S. Yu, K. Ren, W. Lou, "Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multiowner settings," in SecureComm'10, Singapore, 2010.

[7] X. Ma, Y. Zhu, X. Li, "An efficient and secure ridge regression outsourcing scheme in wearable devices," Computers & Electrical Engineering, 2017, DOI: 10.1016/j.compeleceng.2017.07.019.

[8] J. Liu, X. Huang, J. Liu, "Secure sharing of personal health records in cloud computing: ciphertext-policy attribute-based signcryption," Future Gener Comp Sy., vol. 52, pp. 67 - 76, 2015.

[9] P. Scheuermann, M. Ouksel, "Multidimensional B-trees for associative searching in database systems," Inform Syst., vol. 7, no. 2, pp. 123 - 137, 1982.

[10] K. Xue, J. Hong, Y. Xue, D. Wei, N. Yu, P. Hong, "CABE: A New Comparable Attribute-based Encryption Construction with 0-Encoding and 1-Encoding," IEEE Trans Comput., vol. 66, no. 9, pp. 1491 - 1503, 2017.

[11] K. Xue, S. Li, J. Hong, Y. Xue, N. Yu, P. Hong, "Two-Cloud Secure Database for Numeric-Related SQL Range Queries with Privacy Preserving," IEEE Trans Inf Forensics Secur., vol. 12, no. 7, pp. 1596 - 1608, 2017.

[12] R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in CCS'06, Alexandria, VA, 2006.

[13] Y. Zhu, Z. Huang, T. Takagi, "Secure and Controllable k-NN Query over Encrypted Cloud Data with Key Confidentiality," J Parallel Distr Com, vol. 89, no. C, pp. 1 - 12, 2016.

[14] D. Song, D. Wagner, A. Perrig, "Practical techniques for searches on encrypted data," in IEEE S&P'00, Berkeley, CA, 2000.

[15] D. Boneh, G. Crescenzo, R. Ostrovsky, G. Persiano, "Public key encryption with keyword search," in EUROCRYP'04, Interlaken, Switzerland, 2004.

[16] Y. Zhu, Z. Wang, Y. Zhang, "Secure k-NN Query on Encrypted Cloud Data with Limited Key-disclosure and Offline Data Owner," in PAKDD'16, Auckland, New Zealand, 2016.

[17] B. Iyer, S. Mehrotra, E. Mykletun, G. Tsudik, Y. Wu, "A framework for efficient storage security in RDBMS," in EDBT'04, Heraklion, Crete, Greece, 2004.

[18] Q. Liu, C. C. Tan, J. Wu, G. Wang, "Efficient information retrieval for ranked queries in cost-effective cloud environments," in INFOCOM'12, Orlando, FL, 2012.

[19] Y. Zhu, Z. Wang, J. Wang, "Collusion-Resisting Secure Nearest Neighbor Query over Encrypted Data in Cloud," in IWQoS'16, Beijing, China, 2016.

[20] R. Agrawal, J. Kiernan, R. Srikant, Y. Xu, "Order preserving encryption for numeric data," in SIGMOD'04, New York, NY, 2004.

[21] Q. Liu, C. C. Tan, J. Wu, G. Wang, "Cooperative private searching in clouds," J Parallel Distr Com, vol. 72, no. 8, pp. 1019 - 1031, 2012.

[22] A. Boldyreva, N. Chenette, A. O'Neill, "Order-preserving encryption revisited: Improved security analysis and alternative solutions," in CRYPTO'11, Santa Barbara, CA, 2011.

[23] P.FARZANA, A.HARSHAVARDHAN," Integrity Auditing for Outsourced Dynamic Cloud Data with Group User Revocation." International Journal of Computer Engineering in Research Trends., vol.2, no.11, pp. 877-881, 2015.

[24] N. Meghasree, U.Veeresh and Dr.S.Prem Kumar," Multi Cloud Architecture to Provide Data Privacy and Integrity." International Journal of Computer Engineering in Research Trends., vol.2, no.9, pp. 558-564, 2015.

[25] A.Shekinah prema sunaina," Decentralized Fine-grained Access Control scheme for Secure

Cloud Storage data." International Journal of Computer Engineering in Research Trends., vol.2, no.7, pp. 421-424, 2015.

[26] P. Rizwana khatoon and Dr.C.Mohammed Gulzar ," SecCloudPro:A Novel Secure Cloud Storage System for Auditing and Deduplication." International Journal of Computer Engineering in Research Trends., vol.3, no.5, pp. 210-215, 2016.

[27] B.Sameena Begum, P.Ragha Vardhini," Augmented Privacy-Preserving Authentication Protocol by Trusted Third Party in Cloud." International Journal of Computer Engineering in Research Trends., vol.2, no.5, pp. 378-382, 2015.