Preserving Privacy in Outsourced Database

Yung-Wang Lin, Li-Cheng Yang, Luon-Chang Lin, and Yeong-Chin Chen

***Abstract*—The data rapidly increases with time. To deal massive data, or called big data, the idea of database as a service has been proposed. Outsourced database provider offers a lot of computing power and storage area. For organizations, they do not need to build their own infrastructure. It can reduce the cost of consumption. However, the data stored in the third-party provider side, if the third party can not be trusted, sensitive data within organization will have leaked crisis. In order to preserve privacy in organization's database, a number of approaches for preserving privacy have been proposed. Although the database has many security problems should be addressed, such as authentication, integrity, access control, and privacy. Also, studies on database need to consider performance efficiently when data updating. But there are huge problems so that we can not address easily. Hence, this paper mainly discusses the recently proposed approaches for preserving privacy only. We classify and organize approaches, and also discuss these. Approaches are divided into two categories, data encryption and data fragmentation. We introduce in simple common approach, give an illustration, and finally discuss challenges for each approach respectively. In the approach of data encryption, we introduce k-anonymity which is a mainstream solution. In data fragmentation, we introduce clustering which is new idea solution. Finally, we summarize these approaches for preserving privacy and discuss research future works.**

***Index Terms*—Preserving privacy, outsourced database, database security.**

I. Introduction

Recently, the privacy of outsourced databases is a popular research topic. As the rapid development of technology and the convenience of digital content, the data by organization is increasing rapidly. To deal with big data, Hacigumus *et al*. explored a paradigm of database as service in 2002. The third party provides a mechanism to allow their customers to create, store and access their databases at provider end [1]. Using outsourced database can help organization reduce hardware equipment cost, system building, but also reduce cost of the personnel department. However, when the all of data be placed in outsourced database provider, the provider is not trusted, sensitive data may have leaked crisis. Hence, the preserving privacy of database becomes very important issues.

In general, security issues in databases are wide research topics [2]. Database security issues have many difference aspects because scholars focus on different features of database security problems or because they make assumptions about how to create secure database models. According to different features of database, scholars were proposed many security policy including user identification/authorization policy, access control policy, inference policy, accountability policy, audit policy and consistency policy. Also, they were proposed some popular secure models including discretionary access control and mandatory access control. Some mechanisms for sensitive applications, reliable encryption and authentication are designed to protect protocol between client and server when network connection is insecure or can not be trusted. However, Evdokimov *et al.* proposed a new definition for Privacy Homomorphisms (PHs) which is used on database [3]. Their scheme let database can against attacks. But this paper surveys popular solutions for preserving privacy on database systems.

Generally, database service providers in order handle massive data from different users. They will choice framework of distributed environment to build their own basic infrastructure. Since distributed database has advantages for scalability and flexibility. Likewise, the other similar framework of object-oriented database systems is another selection (an object corresponds to a notion of a relational tuple, a row). The outsourced database provider will also use this framework to handle massive data. Above mentioned for different frameworks of database are in order to enhance performance efficiently when the data have changeable in the future. Although the distributed database has a number of security issues need to be addressed, such as access control, confidentiality, reliability, consistency and recovery [4]. Furthermore, existing database security models are not suitable for object-oriented database system because it has wide differences with relational database systems. Millen *et al.* implemented multilevel database system based on object-oriented database using mandatory model [5]. But this paper will focus on privacy issues and discuss it [6]. As organization's own data stored in third-party provider side, if the provider is a malicious adversary, the sensitive data, such as trade secrets, may be stolen by the provider, which is a great threat for organization. For this reason, many solutions for protect privacy on database have been proposed.

We give a scenario which background is a hospital to illustrate the privacy issues in outsourced database. In traditional scenario, the patient's medical records and clinical data will be directly stored which hospital has patient's medical records. However, when that patients change a hospital for medical treatment. There are no patient's medical records in the new hospital. Therefore, the new hospital should investigate the history of medical records of old hospital of the patients, and establish medical records of the

Manuscript received February 11, 2014; revised April 24, 2014.

Yung-Wang Lin and Yeong-Chin Chen are with the Department of Computer Science and Information Engineering, Asia University, Wufeng Taichung Taiwan.

Li-Cheng Yang is with the Department of Management Information Systems, National Chung Hsing University, Taichung, Taiwan.

Luon-Chang Lin is with the Department of Management Information Systems, National Chung Hsing University, and also with the Department of Photonics and Communication Engineering, Asia University, Taichung, Taiwan (e-mail: iclin@nchu.edu.tw).

DOI: 10.7763/IJCCE.2014.V3.350

patient into new hospital. However, when the patients again change a hospital for medical treatment, the new one also needs investigate all histories of medical records that patients visited hospitals in the past. Also, the new hospital needs establish new medical record for that patient. Thus, when the patients change the hospital for medical treatment in the future, the new hospital should investigate all histories of medical records of all old hospitals and establish new medical record for patient. These medical records are hospitals to establish their own, and not synchronized patient records with other hospitals. In order to provide a good medical care for patients, patient's record needs to be synchronized by all hospitals, but also eliminates the inconvenience of survey old records or establish new records. These sorts of inconvenience, these hospitals will be prompted to choose a third-party service provider to store medical records. In the traditional scenario, the patient's medical records are stored directly in different hospitals by different hospitals medical treatment. The new scenario proposed by Fung *et al.* [7], we simply distinguish three different roles, data owners, data publisher, and data recipient. *Data owners* are general patients, such as Alice and Bob. *Data publisher*, a hospital, is responsible for collecting and recording all medical records for patients. *Data recipient* is a service provider which offers a space to store all medical records and clinical data of hospital. The process of data stored in service provider end show in Fig. 1. When the patient visits the hospital, medical records will be collected and recorded into each hospital. Every hospital is a data publisher, they individual provide the recipient of the data to third-party service provider, data recipient. This paper mainly is for privacy issues that the process of data stored from data publisher to data recipient, when data recipient is can not to be trusted, to discussion and organization.

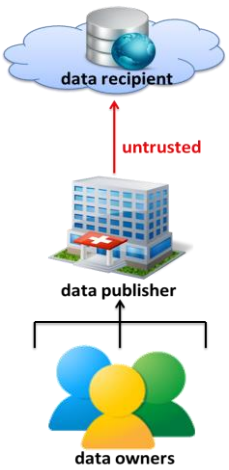

Fig. 1. The process of data stored in outsourced database.

The approaches for preserving privacy divide into two categories by this paper. The first one is data encryption, and the second is data fragmentation. The typical approach for data encryption is encrypts entire databases on the client side before store to the non-trusted third party provider, external database. In order to do query on database, the user needs to download entire database from the third party provider. After, they should decrypt entire database before query it. We can imagine that this approach in practical require many computation time and inefficient. To solve this problem, we can make part of the data encryption, or add some noise in the data. This approach not only can reduce the burden on the system, but also more efficient. The second approach for preserving privacy is data fragmentation. The approach mainly technique is assign data to n blocks and distribute blocks to store in different databases. Give a relation R. A fragmentation of R produces R_1, R_2,…., R_n. Therefore, this method does not require the data for encryption and decryption. Hence, the efficiency of data processing is better than the first approach. The approach for data fragmentation can be divided into two categories. The first is the horizontal partition. Each fragmentation contains a subset of R's rows. This method can prevent the adversary count values for each column, such as the sum or average. It is usually applied in statistical type database. The second one is the vertical partition. This method can prevent the adversary distinguish each entity's value. It can avoid single personal data leakage. The approach applies in protection for information of personal relevance. For example, patient information should be protected in medical database.

The rest of the paper as follows. This paper mainly describes recent researches in the privacy of the database. Approaches of preserving privacy on databases can be divided into two categories: The first one for the data encryption, most approach using k-anonymity, the adversary cannot observe the value in database when using these techniques. The second for the data fragmentation, partition through these technique reach to protection. In the next Section II, we will discuss k-anonymity approach using on privacy protection. Section III, we will discuss data fragmentation, and data partitioning how to using. In addition to the two major technical, some researchers will also combine both of two, which will be discussed in Section V. Finally, in Section VI, we will do a briefly summarize and discuss future works.

II. Data Encryption

In the Section II, we will discuss the encryption approach for preserving privacy in database. In principle, general cryptographic solutions to the problem can be divided to two ways, symmetric or asymmetric encryption schemes. The computing time of symmetric encryption is better than asymmetric. Because it just use single key to encrypt plaintext and decrypt ciphertext. Many proposals are based on symmetric encryption. However, some environment will be promoted by using asymmetric encryption schemes, because symmetric encryption is not security enough. Since asymmetric encryption has more scalability. We can use public key to encrypt plaintext to ciphertext, and decrypt it by using private key. Also, the private key can be used to sign the document, while achieving non-repudiation.

However, most approach for preserving privacy using k-anonymity in recently [8]-[10]. Data displaying through k-anonymity by generalized. Each record in attributes is identical with at least (k-1) other records. This way, adversary can not identify each entity through observation directly. In this section, we will discuss protection data for privacy by approach of k-anonymity.

A. Anonymity by Generalization

Generally, in order to avoid the adversary to directly observe value in the database. We can add some noise into value in the database. In other words, when user queries the database, the database does not answer the correct result to the user. The results are returned to make some modification in order to disguise the correct data. Let $V = \{v_1, v_2, ..., v_n\}$ be a subset of values in attributes *A*, $V \subseteq A$, and values in *V* will mapping to $\overline{V}$. Multiple values correspond to the same value is generalized. Hence, adversary only sees value $\overline{V}$ in attributes, rather than $V_1, V_2, \ldots, V_n$. For an example, assume a database D with two attributes, Age (A1) and ZIP (A2). A record $r = (22, 98003)$ can be revised by generalization to $\overline{r} = ([20-30], [98000-98100])$.

In an attribute with the same value of k is the k-anonymity approach. In other words, records exist in a certain number of indivisible blocks when a user queries the database. Making the attacker can not determine the actual number of sensitive attributes to reach preserving privacy. To meet the k-anonymity of nature, we need to do deal with quasi-identifiers. Quasi-identifiers are a subset of attributes, which can be used to identify an individual. The difference with primary key, primary key is only a single attribute, but quasi-identifiers have one or more attributes. When these attributes merge, it can identify each entity in the relation, such as Name and Age. Let quasi-identifiers $Q = \{A_1, A_2, ... A_n\}$, be a subset of attributes. $\overline{A_1}, \overline{A_2}, ..., \overline{A_n}$ be corresponding subset, and each A_i will mapping to $\overline{A_i}$. Each attribute of A_i has *k* values correspond to the new value $\overline{V}$ by generalization operation, *k*=3. We use Table I to illustrate, suppose there is a simple database there is only one table, for a brief description, and only five attributes, SSN, Name, Age, ZIP, Illness, and Treatment. Which has a quasi-identifiers include Name, Age, ZIP three attributes, $Q = \{\text{Name}, \text{Age}, \text{ZIP}\}$. For a brief description we do not consider is how to produce quasi-identifiers. Table II shows the result after 3-anonymity. All values of attributes in quasi-identifiers has at least three are anonymous. Such as last name of Hellman correspond to Hellman, display last name only, Age 24 correspond to age [20-30], ZIP code 94141 correspond to [94140-94150]. The value is changed using star symbol in front.

TABLE I: PATIENT RELATION

SSN	Name	Age	ZIP	Illness	Treatment
322-42-4224	A. Hellman	20	94141	SARS	Antibiotic
423-22-2522	B. Hellman	27	94142	Diabetes	Insulin
522-93-5221	C. Hellman	24	94143	Flu	Neuraminidase
422-52-5225	D. Mekinley	50	74224	SARS	Antibiotic
942-25-8223	E. Ripley	55	23314	Flu	Neuraminidase
522-92-5221	F. Dooley	43	74224	diabetes	Insulin

TABLE II: THE RELATION AFTER 3-ANONYMITY

SSN	Name	Age	ZIP	Illness	Treatment
322-42-4224	*Hellman	*[20-30]	*[94140-94150]	SARS	Antibiotic
423-22-2522	*Hellman	*[20-30]	*[94140-94150]	Diabetes	Insulin
522-93-5221	*Hellman	*[20-30]	*[94140-94150]	Flu	Neuraminidase
422-52-5225	D. Mekinley	50	74224	SARS	Antibiotic
942-25-8223	E. Ripley	55	23314	Flu	Neuraminidase
522-92-5221	F. Dooley	43	74224	diabetes	Insulin

B. Challenge

Since individual sensitive data is easily inferred by an attacker. But if the numbers of values are combined into one and the same value or range. It does not have the sensitivity. Patients age are changed to a range 20-30 is show in Table II. The attacker can not determine the actual age of the patient. There is no way to infer individual patient using merging Name, Age, ZIP code. The advantage of k-anonymity is resisting the attacker's inference attack. However, the database reply to a correct value is a problem when users query anonymous database. Because this approach will revises original value to another incorrect value. In other words, the original value will be broken by this approach. For example, a doctor queries actual age of the patient who SSN is 322-42-4224, such as Table II. Database needs to be restored to their age range to a single value. Reducing the error information is a challenge for this study. However, some specific databases, such as numerical database or statistical database, are suitable to use this approach because they can allow a small amount of error. Besides, as time increases the data in the database will be grown. When updating data, how to keeping incremental data privacy on k-anonymity is another challenge. Most existing approaches address this problem via re-anonymizing all data in entire database. Zhang *et al.* [11] proposed an efficiently approach to address problem of incremental data in 2013. It is another idea through indexing quasi-identifiers to dynamical maintains existing anonymized data and keeps k-anonymity when updating data with new data. Consequently, privacy preservation over incremental data is still challenging in incremental database. In addition, the optimized k-anonymity is a NP-hard. Give a relation R which has a set of attributes $A_1, A_2, \ldots, A_n$. Each attribute will correspond into new attributes $\overline{A_1}, \overline{A_2}, ..., \overline{A_n}$. It should be considered an attribute with *k* values are anonymous. *k* is defined quite important. It will be multiple records to be anonymous if *k* is too large. That would enhance the system calculates time. On the contrary, it will reduce privacy degree if *k* is too small. Find out the optimal k-anonymity algorithm and minimal loss of information is other challenge for this approach [9].

III. DATA FRAGMENTATION

Since the system will reduce performance when using approach for data encryption. Hence, Aggarwal *et al.* proposed a new idea is assign two data storages to store entire database. And two storages can not communicate with each other [12]. In the database, some data attributes merger will leak individual privacy. Assuming a combination of name and birthday can distinguish individuals. These attributes are sensitive attribute. Its collection is called quasi-identifiers. Main purpose for data fragmentation is to separate the sensitive data and store in different databases. When the data is distributed in different databases, adversary can not understand the record in databases. A single data does not make sense, if there is only a data set of age in a single database. Illustrated in Fig. 2, give a relation *R*, and *R* can be partition to *n* fragments $f_1, f_2, \ldots, f_n$.

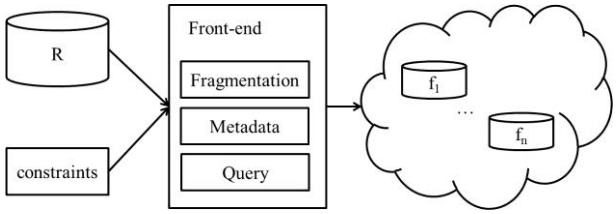

Fig. 2. Overview of data fragmentation.

$C_0: \{SSN\}$
$C_1: \{Name, Age\}$
$C_2: \{Name, Illness\}$
$C_3: \{Illness, Treatment\}$
$C_4: \{Age, ZIP, Illness\}$
Fig. 3. Confidentiality constraints.

A. Confidentiality Constraints

Data fragmentation is based on confidentiality constraints. Constraints will clearly define the merger of those attributes will leak privacy. It can easily achieve the purpose of preserving privacy when we only need to separate these attributes to different database storages. It is a pre-processing for constraints definition before data partition. We need find all the quasi-identifiers sets in the database. Then we can know the merger of attributes whether safe by the quasi-identifiers. It is an important task to find constraints. The algorithm for automatic detection of Quasi-identifier be proposed in 2010 [13]. This in practice is given to each attribute a weight which value higher is meaning more sensitive. Based on the weight, we can calculate a value for each attribute the probability of emergence. Let attributes which is higher than user defined threshold be a collection. And then we can define all constraints that are meaning each collection is a constraint. Constraints should be satisfied follows. Let $C = \{c_1, c_2, ...c_n\}$ is a set of constraints, and Q is a set of quasi-identifiers, $c \subseteq Q$, $\forall c_i, c_j \in C$, $c_i \not\subset c_j$.

B. Data Partition Using Clustering

Data partition is based on the above mentioned constraints, and then assigns sensitive data to different areas. The main technique for data partition is sensitive data will distribute in different part of the storage. Let a relation *R*, and fragments $F = \{f_1, f_2, ..., f_n\}$ are partition from *R*, $F \subseteq R$. The approaches can be divided into two categories. The first is the horizontal partition which is a set of columns. The second is the vertical partition which is a set of rows. Recently approach is based on clustering attributes [14]. We simply explain to data partition approach which is based on clustering. For an example, assume that an original relation with Table I. According to the constraints above mentioned, we designed five constraints shown in Fig. 3. According C0 SSN is a singleton of constraint. The SSN itself will leak privacy. Hence, we do not consider data partition for SSN because it can use traditional cryptographic scheme to address. According to C1, the merger of Name and Age will leak privacy. So that two attributes need to be placed in two different databases, such as Name be stored in database A, Age be stored in database B. According to C2, the merger of Name and Illness will leak privacy. So that two attributes should be placed in two different areas. But due to a combination of Age and Illness does not leak privacy. So that Illness can be stored in B. According to C3, Illness and Treatment must not be placed in the same database. But the Treatment and the Name can be stored together. So the Treatment is stored in A. According to C4, a combination for Age, ZIP and Illness are insecurity. Age is stored in B by C1. Illness is stored in B by C2. Hence we will store ZIP to A. When these three attributes can not be combined to a merger, it can protect the privacy of the database. Finally we can be divided into two clusters. In order to preserving privacy, the approach for data fragmentation is to distribute sensitive data and store in different places.

C. Challenge

The approach for data fragmentation does not affect the system operation time. It will not cause the database burden. It is a good way for distributing sensitive data in different areas. The main problem for data fragmentation is determining the granularity of data blocks. Sensitive data may still be placed together if the granularity is too big. It has information leak crisis. On the contrary, it will reduce system efficiency using query if granularity is too small. When a user queries the database, front-end needs more time to combine the different block, and accurate back to the user. The response time after user do action of query should be real time. Hence, query processing is this approach need be face challenge. Usually we wish to design optimal data fragmentation, minimal fragmentation. But when each granularity of fragment is different to others, it is a knapsack problem, NP-hard. We are really difficult to find an optimal data fragmentation algorithm. Additionally, since data are stored in different areas. We also need to build an index table to record location for data. We can find storage location based on index table when a user quires the database. Also we can merge the correct data back to the user. How to efficiently query to this approach is a challenge. Further, although this approach can resist external attacks. When there is a legitimate user but is a malicious attacker. Front-end protection is quite important. Because storage location and query methods are built on the front-end. When the front-end crushed by attacker, an attacker can obtain sensitive data. In addition, approaches of data fragmentation will face the problem of incremental data in the future. In some environment, such as bank, data transaction is a daily work. To keep data has high assurance, availability, performance, and scalability, relating algorithm of dynamic allocation, has been proposed [15]. We should consider location for new data and existing data, and dynamic allocate file to corresponding area, and keep performance efficiently. Section summarily, data granularity determine, efficiency query, protection for front-end, and distributed algorithm for file allocation are challenges for this approaches.

IV. Hybrid of Encryption and Fragmentation

There are two approaches for preserving privacy on database. One is encrypted, and the other is data fragmentation. These two kinds of privacy protection is a double-edged sword. Although encryption approach can enhances degree of privacy. Contrast, it need for encryption and decryption operations. The efficiency for database

operation will reduce. Another approach is for data fragmentation. There is no encryption and decryption of the time, the efficiency is higher. But the data stored plain text data. The privacy degree is lower than first one.

The choice of these two methods is a trade-off. Therefore, in order to strike a balance between the two. Some of the approach will combine both of two [16], [17]. We briefly describe the approach for hybrid encryption and data fragmentation. First, we should find out all the quasi-identifiers, sensitive attributes. Based on quasi-identifiers defines the constraints. This is a pre-processing for data fragmentation. Followed by sensitive attributes can not be placed in the same database as the principle, we can separate the original relation R to $f_1, f_2,\ldots, f_n$, So far we only use the data fragmentation approach. Finally, we find the minimal fragment to be encrypted, such as DES, AES, etc., or simply use the k-anonymity. The Fig. 4 illustrates hybrid approach. User data will be partition to n fragments, $f_1, f_2,\ldots, f_n$, by fragmentation process, and then each fragment will be encrypted by encryption process. The Front-end responses results through query process merge data from different fragments if user query database.

In order to reach optimal algorithms, hybrid approach adopts advantage of high privacy for encryption and efficiency for data fragmentation. This is different view point to keep privacy on outsourced database. This approach is to obtain the advantages of the two above approaches. It is difficult to design optimal algorithm in practical for both of two approaches. The choice of two is a trade-off. Some researchers combine both of two to design nearly optimal algorithm.

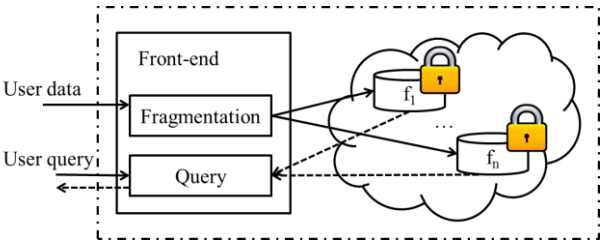

Fig. 4. Hybrid of encryption and data fragmentation.

V. Conclusion

This paper is mainly surveys recent research for database privacy, and to discuss these approaches may face challenges. We divide these approaches into two categories. One is a typical approach for data encryption. We clearly understand approach for k-anonymity which is the most approach for preserving privacy. Hence, it is often being applied in the protection of the privacy of the database. Meanwhile, we also discussed the approach for encryption will reduce efficiency on database operation. But it is feasible for numerical database and statistical database. The other one is data fragmentation. In this approach, we discussed recently methods are based on clustering for data fragmentation which is another choice for preserving privacy on distributed database. We also discussed optimal algorithm for fragment is a NP-hard, so we just only can design an algorithm for approaching optimal. Besides, query processing should be considered after data partition process. Finally, since the choice of these two approaches is a trade-off. Hence, there is a new idea of hybrid approach of encryption and fragmentation has been proposed. We briefly discussed this approach. This is to balance these two approaches, hoping to find a best way.

As data continue to increase, we would need to do the actions for insert, delete, update, and query on database. Preserving privacy on database is increasingly important. However, although there are many approaches have been proposed to preserving privacy. The protection for privacy is another issue when the data is constantly updated. If we are using k-anonymity approach, we should keep each record is identical with at least (k-1) other records on database. If we are using data fragmentation approach, we should consider whether re-partition while avoiding excessive partition. So how to keep privacy when database updating that is we can extend to study.

References

[1] H. Hacigumus, B. Iyer, and S. Mehrotra, "Providing database as a service," in *Proc. IEEE International Conference on Data Engineering*, 2002, pp. 29-38.

[2] S. Imran and I. Hyder, "Security issues in databases," in *Proc. IEEE International Conference on Future Information Technology and Management Engineering*, 2009, pp. 541-545.

[3] S. Evdokimov, M. Fischmann, and O. Gunther, "Provable security for outsourcing database operations," in *Proc. IEEE International Conference on Data Engineering*, 2006, pp. 117.

[4] Z. S. Zubi, "On distributed database security aspects," in *Proc. IEEE International Conference on Multimedia Computing and Systems*, 2009, pp. 231-235.

[5] J. K. Millen and T. F. Lunt, "Security for object-oriented database systems," in *Proc. IEEE Symposium on Security and Privacy*, 1992, pp. 260-272.

[6] P. Samarati and S. Vimercati, "Data protection in outsourcing scenarios: Issues and directions," in *Proc. ACM Symposium on Information, Computer and Communications Security*, 2010, pp. 1-14.

[7] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *Journal of ACM Computing Surveys*, vol. 42, no. 4, pp. 14:1-53, 2010.

[8] G. Aggarwal, T. Feder, and K. Kenthapadi, "Anonymizing tables," in *Proc. International Conference on Database Theory*, 2005, pp. 246-258.

[9] R. Bayardo and R. Agrawal, "Data privacy through optimal k-Anonymiztion," in *Proc. IEEE International Conference on Data Engineering*, 2005, pp. 217-228.

[10] T. Tassa and E. Gudes, "Secure distributed computation of anonymized views of shared databases," *Journal of ACM Transactions on Database Systems*, vol. 37, no. 2, pp. 1-43, 2012.

[11] X. Zhang, C. Liu, S. Nepal, and J. Chen, "An efficient quasi-identifier index based approach for privacy preservation over incremental data sets on cloud," *Journal of Computer and System Sciences*, vol. 79, no. 5, pp. 542-555, 2013.

[12] G. Aggarwal, M. Bawa, and P. Ganesan, "Two can keep a secret: A distributed architecture for secure database services," in *Proc. Conference on Innovative Data Systems Research*, 2005.

[13] Y. Yu, "Privacy protection in secure database service," in *Proc. IEEE International Conference on Networks Security, Wireless Communications and Trusted Computing*, 2010, pp. 218-222.

[14] T. J. V. R. K. M. K. Sayi, R. K. N. S. Krishna, R. Mukkamala, and P. K. Baruah, "Preserving privacy of outsourced data: A cluster-based approach," in *Proc. IEEE International Conference on Information Reuse and Integration*, 2012, pp. 215-223.

[15] A. Mei, L. V. Mancini, and S. Jajodia, "Secure dynamic fragment and replica allocation in large-scale distributed file systems," *Journal of IEEE Transactions on Parallel and Distributed Systems*, vol. 14, no. 9, pp. 885-896, 2003.

[16] V. Ciriani, S. Vimercati, S. Foresti, and S. Jajodia, "Combining fragmentation and encryption to protect privacy in data storage," *Journal of ACM Transactions on Information and System Security*, vol. 13, no. 3, pp. 1-33, 2010.

[17] V. Ciriani, S. D. C. D. Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Fragmentation and encryption to enforce privacy in data storage," in *Proc. European Symposium on Research in Computer Security*, 2007, pp. 171-186.

Yung-Wang Lin is a PhD student in Department of Computer Science and Information Engineering, Asia University, Wufeng Taichung Taiwan. His current research interests include information security and cloud computing. He also employed in NanKai University of Technology.

Li-Cheng Yang is a graduate student in Department of Management Information Systems, National Chung Hsing University, Taichung, Taiwan. His research interests include data hiding, cryptography, privacy protection, distributed database, and nosql framework. He intends to expand research issues to big data and embedded systems in the future. He also attends activities of open source community, such as TCFFM, COSCUP 2013, MOPCON 2013.

Iuon-Chang Lin received the Ph.D. in computer science and information engineering in March 2004 from National Chung Cheng University, Chiayi, Taiwan. He is currently a professor of the Department of Management Information Systems, National Chung Hsing University, Taichung, Taiwan. His current research interests include electronic commerce, information security, cryptography, and mobile communications.

Yeong-Chin Chen received the M.S. and Ph.D. degrees in electrical engineering in 1989 and 1998 respectively, from Cheng Kung University, Taiwan. He joined the Marine Science and Technology Center of Chung Shan Institute of Science and Technology, Taiwan from 1989 to 2000, where he was a senior researcher in the Underwater Technology Department. He is currently a professor in computer science and information engineering at Asia University, Taiwan. His research interests include acoustical transducer engineering, power signal measurement & anaysis, and software engineering.