Using Multi Shares for Ensuring Privacy in Database-as-a-Service

Mohammed Abdullatif ALzain and Eric Pardede.
Department of Computer Science and Computer Engineering,
La Trobe University, Bundoora, Australia.
Email: [maalzain@student., E.Pardede@]latrobe.edu.au

Abstract

Database-as-a-service (DAAS) is a new model for data management, where a service provider offers customers software management functionalities as well as the use of expensive hardware. This service enables data integration and access on a large scale in cloud computing infrastructures.

Addressing data privacy in DAAS is considered a significant issue for any organizational database. Due to the fact that data will be shared with a third party, an un-trusted server is dangerous and unsafe for the user. This paper proposes the architecture of a new model appropriate for NetDB2 architecture, known as NetDB2 Multi-Shares (NetDB2-MS). It is based on multi-service providers and a secret sharing algorithm instead of encryption, which is used by the existing NetDB2 service. The evaluation is done through simulations. It shows a significant improvement in performance for data storage and retrieval for various query types.

1. Introduction

In any organization, the use of technology to store, evaluate, and process data in databases is extremely significant. Consequently, organizations have to translate data into meaningful information to help them improve their decision-making or create strategic advantages [14]. In addition, complete and correct data plays a major role in customer relation management or internal resolution [16]. As a result, the high quality of data in a database is a significant issue for any business.

Data processing is an important data management issue for any organization. Efficient data processing leads to efficient results from the data. A traditional solution for the company is to purchase software and hardware, hire database professionals, and establish network connectivity to manage their data. This solution has become increasingly expensive and complicated [10].

Recently, for the new generation in the commercial world, various computing needs are provided as a service. The service providers take care of the customer's needs by, for example, maintaining software or purchasing expensive hardware. For instance, the service EC2, created by Amazon provides customers with scalable servers. As another example, under the NSF CLuE program, NSF joined with Google and IBM to offer academic institutions access to a large-scale distributed infrastructure [1].

All companies need to be able to restore data, back up data, and reorganize data to reclaim space [10]. However, an available solution to transferring a database from one version to another without adversely impacting on it is still in its infancy [9].

Storage and retrieval of data are considered to be big challenges, especially for small companies because of the high cost involved. The data management cost is estimated to be between 5 to 10 times higher than the data gain cost [11]. In addition, when dealing with in-house data management, it is important to have a set of higher level skills to deal with issues such as disaster recovery, storage technologies, capacity planning, and DBMS and Operating Systems software upgrades. Most organizations would prefer to use their technical resources to focus on their local business applications instead of becoming a data management company [1].

In relation to the above, a database-as-a-service (DAAS) provider is the solution. Authors in [10] claim that DAAS provides the instrument for storing, accessing and creating databases. Furthermore, they reveal that customers or users can now benefit from DAAS by accessing their data via the database service provider's hardware and software instead of using their own company's infrastructure. Therefore, the user should not be adversely affected by network or hardware failure or changes in the database service provider's site. However, the user should use only a ready and maintained system offered by the service provider.

The practice of DAAS is essential to facilitate cloud computing infrastructures. In cloud computing, a large amount of data is integrated and used by various

1530-1605/11 $26.00 © 2011 IEEE

parties through the internet. Such integration, however, raises the issues of data privacy and security. DAAS users want to maintain the privacy of their data even though the data is available in the cloud through the service providers.

In this paper, we focus on the data privacy aspect of DAAS. Due to the fact that data and information will be shared with a third party, an un-trusted server is dangerous and unsafe for the user. Protecting private and important information such as credit card details or patients' diseases is significantly important, especially from attackers or database service providers [15]. The purpose of this paper is to provide a new model which ensures privacy in DAAS. The remainder of this paper is organized as follows. Section 2 summarizes previous work in the area of DAAS and its privacy issues. Section 3 introduces the proposed model called NetDB2 Multi-Shares, with a thorough data flow explanation. Section 4 describes the implementation and evaluation, where we compare the newly proposed model with existing NetDB2. Section 5 presents the conclusion and the future work.

2. Related works

2.1. Database-as-a-service: preliminary

According to [1], DAAS is a new data management model in which a third party service provider hosts a database and provides the related software and hardware to support data for organizations. The service providers perform various functionalities such as database backup, restoration, administration, and reorganization to reclaim space for data, data transfer from one version to another, and other data management operations [5].

The benefits of the DAAS provider as evaluated in [1] are as follows: the provider provides data management for its customers, thereby negating the need for the customer to purchase expensive software or hardware; customers do not need to hire professionals to administer or maintain their databases or upgrades of software; the provider offers data storage at a low cost; and customers are able to access their data from their own location or home through the internet in a private and secure way instead of carrying data as they travel, leading to a reduced risk of data loss or theft.

There are two common types of DAAS. The first type provides data and service at the same time. The second type provides service only.

2.2. Challenges for DAAS

The main challenge in DAAS research is to build a healthy and efficient service for securely managing and preserving data [1]. The provider needs to guarantee that data and the query results are secure and that the queries can be executed efficiently and correctly. Therefore, one of the main challenges for the service provider is data privacy [1],[10]. This has been backed up by our previous literature survey study provided in Table 1. This table shows that privacy is an integral part of data quality research in DAAS, and it has received much research interest in recent years.

			Quality Factors for DAAS			
Ref	**Year**	**DAAS**	**Data Privacy /Security**	**Data Performance**	**User interface**	**Privacy/ Security Mechanism**
[1]	2009	√	√			Multi Sharing
[15]	2009	√		√		
[18]	2008	√	√			Encryption
[5]	2008	√		√		
[4]	2007		√			Ghost DB
[21]	2007	√	√			Assurance 3 factors
[22]	2007		√			
[20]	2005	√	√			Cryptographic proof mechanism
[3]	2004		√			Encryption
[13]	2004	√	√	√		Data Partitioning (Bucketization)
[2]	2003		√			Information Sharing + Developed Protocol
[10]	2002	√	√	√	√	Encryption (Blowfish)
[11]	2002	√	√			Encryption
[6]	1998		√			Method for data retrieving
[19]	1979		√			secret sharing method

Table 1. Related Work on DAAS Data Quality

In this section, different points of view about data privacy will be discussed. The aim is to build a healthy and efficient service for managing data in a secured way, ensuring that even if the data is outsourced to a third party, the individual data values will not be visible to the service provider [11],[3],[4],] [13].

If something happens to the data or if the data is corrupted by the service provider, the service provider is responsible for data restoration [1]. The service provider should have a mechanism to recover or back-up the data.

There are three issues to be addressed for a wide adaptation of the DAAS framework in terms of data

privacy and security: *Encryption-based Data Security, Private Information Retrieval, and Information Distribution* [1].

2.2.1. Encryption based data security. Most proposed approaches for secured DAAS are based on data encryption to hide the content of the data from service providers [3],[11]. This approach however, increases the response query time due to the computational complexity of data encryption/ decryption during a query execution. Computing a privacy-preserving intersection problem using encryption results in a very high time cost [2].

A model for the cost problem regarding data encryption is provided in [1]. The estimated cost of data encryption for 10 documents (each with 1,000 words) at one site and for 100 documents at another site might take 2 hours of computation and about 3 GB of data transmission. Similarly, for a real database of 1 million medical records, data encryption can take about 4 hours of computation time and 8 GB of data transmission.

Another problem with data encryption is finding the required tuples to execute the query over encrypted data. In order to solve this problem, the authors in [11] and [13] proposed a filtering mechanism that selects required tuples depending on the data content. A better filtration mechanism results in lower communication cost for retrieving data from the service provider. Overall, it will impact on the query response time in a positive way. However, the quality of the filtration process is dependent on the amount of information presented to the service provider. Hence, there is a privacy performance exchange in the above solutions [1].

2.2.2. Private information retrieval. The problem of private information retrieval has been a research topic for more than a decade. This problem was proposed for the first time in the context of a user accessing third-party data without referring to the third-party interests [6]. For instance, if a customer wants to retrieve information from a database made available by a company, this customer would not like his/her future intentions to be visible to that company. This consideration is related to the context of outsourced data. A customer may not want to send his/her query to the service provider unless this information remains private [1].

There is a lengthy history of theoretical research in the area of obtaining sub-linear communication complexity. One issue is the availability of multiple service providers. Repeating the database at several servers may solve this problem [6],[22]. The concept of private information retrieval which ensures the privacy of user queries has also been extended to include the privacy of data. This is referred to as symmetric private information retrieval [17], [8].

2.2.3. Information distribution. Authors in [1] discussed the issue of information distribution with the aim of showing that there is an orthogonal approach which is based on information distribution instead of encryption in the area of data and computer security. The need to communicate important or private information from one party to another instigated most of the work on data security.

Authors in [1] introduced Shamir's secret sharing algorithm [19] as a solution for the privacy issue. In [19], the authors proposed dividing the data D into (n) pieces ($D_1....D_n$) in such a way that knowledge of any k or more D_i pieces makes the value of D known. Therefore, a complete knowledge of $(k - 1)$ pieces reveals no information about D, k should be less than n to keep the value of shares un-constructible and ensure that the adversary cannot access k data pieces. Shamir's method theoretically secures information

In addition, by using a (k,n) threshold scheme with $n = 2k – 1$, the authors in [1] can have a strong key management scheme. The goal is to take a distributed approach to secure DAAS [1], the reason being that they want to explore the use of a secret-sharing approach and multiple service providers. The advantage of this approach is that it addresses both privacy-preserving querying and the data security of outsourced data.

2.2.4. Overview of the secret-sharing approach. Instead of encryption, the secret-sharing method [1] distributes data to multiple servers to ensure privacy of user queries. If the client wants to outsource the data from data source D to database service providers ($DAS_1,\ldots,DAS_n$), data should be divided into n shares and each share should be stored in a different DAS. After this, the written query in D will be sent to all DASs to retrieve the relevant shares to answer a client query without knowing the value of data shares from the service provider.

In order to reconstruct the secret value v_s at the data source, the knowledge of any k can refer to v_s besides some secret information X that is known only to the data source. Therefore, even with a full knowledge of $(k – 1)$, DAS will not have any knowledge of v_s even if X is known to them [16]. Furthermore, D chooses a random polynomial $q(x)$ of degree $(k – 1)$, where the constant is v_s. Each DAS has v_s as well as X which is a set of n random points. Figure 1 illustrates how the shares can be written [1].

$$\text{Shares } (v_s, 1) = q(x_1) = ax_1^{k-1} + bx_1^{k-2} \ldots + v_s$$
$$\text{Shares } (v_s, 2) = q(x_2) = ax_2^{k-1} + bx_2^{k-2} \ldots + v_s$$
$$\vdots$$
$$\text{Shares } (v_s, n) = q(x_n) = ax_n^{k-1} + bx_n^{k-2} \ldots + v_s$$

Figure 1. Example of Shares using Polynomial Function [1].

Figure 2 shows an example of data source *D* that has *EMPLOYEE* table and *SALARY* attribute. *D* wants to outsource the *SALARY* to three DASs (DAS_1, DAS_2, and DAS_3). In addition, *D* chose 5 random polynomials with degree one (this is for each SALARY). Since the number of shares $n = 3$, $k = 2$ and the secret information X $\{x_1=2, x_2=4, \text{ and } x_3=1\}$ will be used by the polynomial functions for distribution to each DAS.

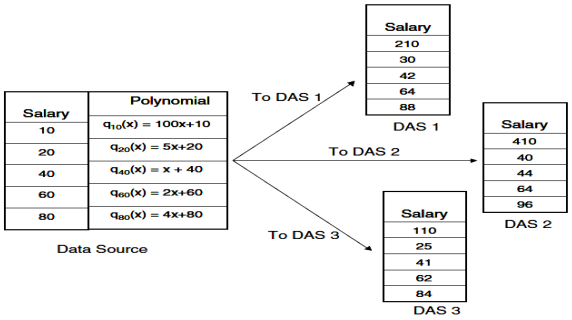

Figure 2. Data Distribution in Multiple DASs [1].

[1] Extends the approach to retrieve the required data from the service provider instead of retrieving all the shares and thus decreases the cost of communication and computation in the processing of the query. They achieved this by constructing order-preserving polynomials with polynomial functions and sending them to each DAS to retrieve any value later according to its order.

2.3. NetDB2 service

NetDB2 is an example of DAAS that has been used by many universities in different locations for teaching database courses, due to its useful tools which allow the creation or removal of tables and the performing of queries through the internet [10].

The data in NetDB2 is located at the server site and the user connects to it through the internet and performs query and tasks through an API such as JDBC or through the web browser [12] (see Figure 3). The goal is to make the system easy for the user without the need for installation or configuration on the user site. The system is portable and available from any location because the users can access it through the web browser [10]. This UI differentiates NetDB2 from other database servers and makes the application easier to use.

2.3.1. Data privacy in NetDB2 service. [10] considered data privacy in NetDB2 in relation to two issues: privacy of data during transmission and privacy of stored data. This paper claims that sufficient studies have been done on the privacy of data during network transmission in terms of the internet, having been addressed by the Transport Layer Security (TSL) protocol and the Secure Socket Layer protocol (SSL) [7], whereas the issue of stored data privacy in DAAS has received less attention [10].

In DAAS, the service provider needs to devise a method to protect the privacy of the customer's data. The service provider needs to provide security measures to ensure that if the data is stolen, the adversary cannot use it for any purpose. Authors in [10] suggest data encryption as an effective technique to solve this problem.

2.3.2. NetDB2 service encryption response time. The researchers compared the response time between two queries against two types of data. The first query was on decrypted data and the other was on encrypted data with the same data overall. The results showed that the response time for encrypted data increased because of both the routine or hardware invocations and the cost of decryption in DB2.

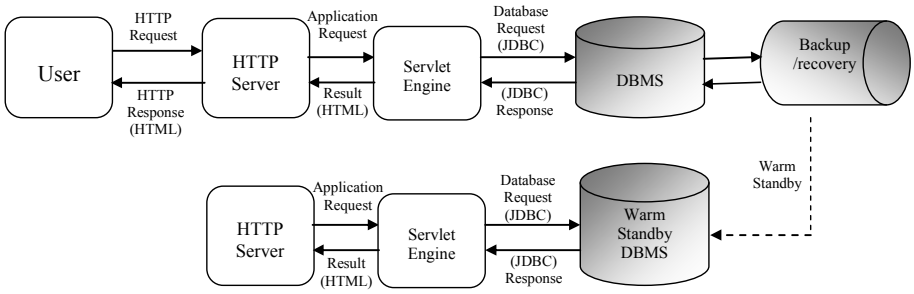

Figure 3. NetDB2 System Architecture [10].

3. Proposed DAAS model

In this section, we propose a new model applied for NetDB2 architecture, called NetDB2 Multi-Shares (NetDB2-MS). NetDB2-MS ensures privacy in DAAS and is based on multi-service providers and the secret sharing algorithm instead of encryption, which was used by the existing NetDB2 service. The purpose of the proposed new model is to avoid the high cost of data encryption and decryption.

We emphasize that the security of data transfer through networks is not the aim of this paper. The security of networks or layers will not be modified in the proposed approach due to the fact that it has been addressed in the previous NetDB2 model.

3.1. NetDB2-multi shares model (NetDB2-MS)

Figure 4 illustrates the general DAAS model. Part A represents the client side, which sends data inquiries to the server in part B. The data source in part C stores the data on the server side. The server side, which is supposed to be a trusted server, ensures the privacy of any client query. A problem occurs if we cannot guarantee that the server is a trusted service. In a lot of research that has used encryption techniques to protect data from an un-trusted server, the encryption and the decryption process takes place at the data source in part C.

The proposed NetDB2-MS model (see Figure 5) does not preserve privacy by encryption; rather, privacy is preserved by using the secret sharing method. This avoids the negative effects of encryption on queries. It focuses on the procedures between the data source and the service providers, not between the client and the http server.

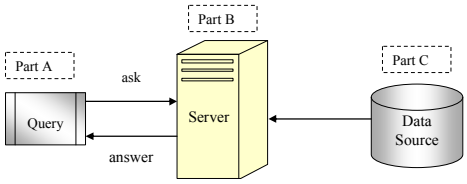

Figure 4. Client/Server DAAS Model.

As mentioned in the previous section, the splitting of data in order to store it in different service providers is considered the main factor in the secret sharing method. There is a lengthy history of theoretical research in the area of obtaining sub-linear communication complexity. One issue is the availability of multiple service providers.

NetDB2-MS consists of three layers: the presentation layer, the application layer, and the data management layer.

The presentation layer contains the end user's browser and HTTP server. The management layer consists of the Database Management System (DBMS) and the database service provider. DBMS communicates with the Servlet Engine through the JDBC protocol. Communication between components is through a secured private high speed network that uses secure protocols.

3.2. NetDB2-MS model data flow

This section considers in depth the data flow in the proposed model and how the user's query can be run through the model in a private and secure way. In addition, it describes how DBMS manages the data, divides them into shares, and distributes shares to separate servers.

3.2.1. Sending data procedure. As described in Figure 5, a user sends a query by using a UI and a web browser through an HTTP request. The HTTP server plays a major role in communication between the web browser and the application. The user's query will be sent from the HTTP server to a Servlet Engine by an application request. The communication between Servlet Engine and the DBMS is done through JDBC protocol.

When the query arrives at the data source, the DBMS will manage the query and send it to the service provider. After the result of the query is returned to the DBMS, it returns the result to the Servlet Engine and then the HTTP server returns the result of the query. All communication between layers and components are done through a secured and safe network.

3.2.2. Procedure between DBMS and DAS. DBMS divides the data into *n* shares and stores each share in a different service provider DAS (see component *A* in Figure 5). The DBMS generates a random polynomials function in the same degree for each value of the valuable attribute that the client wants to hide from the un-trusted servers. The polynomials are not stored at the data source but are generated when the query is received from the user at DBMS.

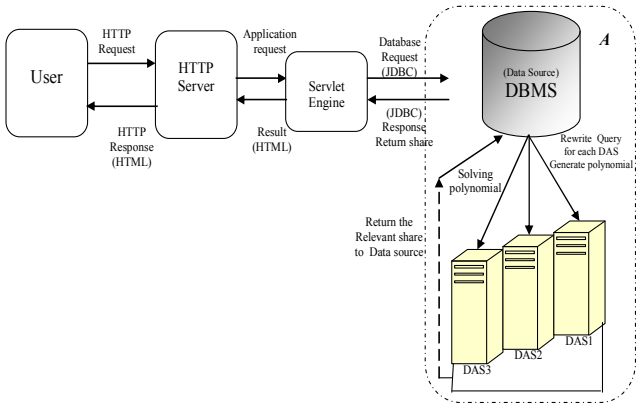

Figure 5. NetDB2-MS Architecture.

When a user's query arrives at the DBMS, the DBMS rewrites *n* queries, one for each service provider and the relevant share will be retrieved from DAS. For instance, the rewritten query for DAS1 retrieves all students whose age is *share(22,1)* where the secret value is the age 22 and the service provider is DAS_1. To find *share(22,1)*, the data source *D* first generates polynomials for the secret value age 22 and the position for the value in the share $p22(x_i)$.

After retrieving the relevant tuple from the DAS, *D* computes the secret value to send to the client through the secured and private network. The secret sharing method can be applied to execute exact match, range, and aggregation query types.

4. Implementation and evaluation

This section explains the secret sharing model and the procedure between DBMS and DAS in NetDB2-MS. It will also present the experimentation that compares the newly proposed model with the existing NetDB2, which uses the blowfish encryption technique.

The goal of this experiment is to show that the multi-sharing secret technique in NetDB2-MS outperforms encryption techniques in the existing NetDB2 service in terms of time performance.

4.1. NetDB2-MS scenario

DBMS divides the data that the user wants to hide from the un-trusted server into *n* shares. After dividing the data (assuming the data is a numeric value, for example, *student ages*) into 4 shares and storing them in different DASs, the DBMS generates random polynomial functions with degree at the same level, one for each student's age in the *STUDENT* table with the actual age as the constant part of the function. These values will then be stored in different DAS providers.

For this scenario, the value of $n = 4$ and $k = 3$. In addition, the DBMS uses the secret information X values (x_1=2, x_2=1, x_3=4, x4=3) to create the secret value. The polynomial for ages {22, 25, 29, 33, and 32} would be: q22(x) = 100x +22; q25(x) = 5x +25; q29(x) = x +29; q33(x) = 2x +33; and q32(x) = 4x +32. Please note that we use the same polynomial functions as in Figure 2.

If x_1 is applied in polynomials, the value of age 22 will be stored as 222 at DAS_1 and stored as 35 at DAS_2 and so on. At this stage, the user's query should have arrived at the DBMS and DBMS should rewrite the query again to retrieve the result from the relevant share from DAS.

After retrieving the relevant value from DAS, DBMS computes the secret value to send it to the client through a secured and private network.

The numeric attribute data type is considered in the secret sharing approach. Authors in [1] briefly discuss how to represent a non-numeric attribute data type. They converted the non-numeric attribute into a numeric attribute to apply a converted attribute to their schema. Any word consists of 27 possible characters which are enumerated (∗=0, A=1, B=2, C=3..., Z=26).

In our scenario, if the user wants to query the address for a certain student living in 'Bundoora', the value of the address will be converted to a numeric value as (12302211441515181) and will execute the polynomial functions on this value before it is stored in DASs.

4.2. Experimentations and evaluation

The experiment is written in Java to simulate data storing in different DAS providers, data retrieval from different DAS providers, data encryption and decryption.

The experiment provides a comparative evaluation

between the secret sharing method that is used in NetDB2-MS and the blowfish encryption technique that is used in the NetDB2 service. The comparison includes operations on data storing and data retrieval. We also present a comparison of all three types of queries, namely exact matching, range and aggregate query.

For the dataset, we use numeric data. Nevertheless, we can still apply this experiment to non-numeral data as it was explained in Section 4.1.

4.2.1 Data storing procedure. Data storing in NetDB2-MS involves data distribution from the data source to different DAS providers. This is done after executing the polynomial functions on the data. On the other hand, the data storing procedure in the NetDB2 service focuses on data encryption before it has been stored.

We compare data storing time between the secret sharing algorithm in NetDB2-MS and the blowfish algorithm in the NetDB2 service with various data sizes in several cycles. Figure 6 shows that the blowfish algorithm outperforms the secret sharing algorithm when we increase the data size. Although blowfish algorithm is faster than secret sharing algorithm, but typically the data will be distributed once to multiple shares. This does not affect our contribution for ensuring the privacy of users' queries during the data retrieval procedure.

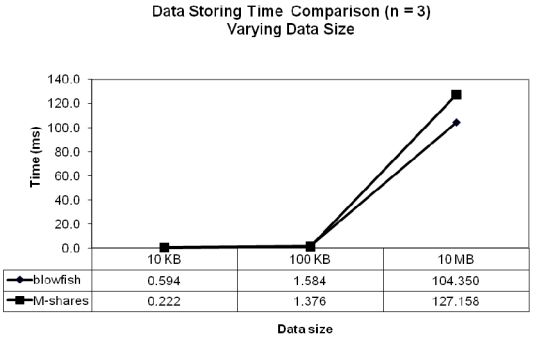

Figure 6. Data Storing Time Comparison, Varying Data Size.

To analyse the effect of a number of shares in our proposed technique, we also perform experimentation for data storing in NetDB2-MS using static data size (10 MB). Figure 7 shows that the time cost for the data storing procedure increases with the number of shares.

Even though the time cost is increased along with the increased number of shares, increasing the number of shares will improve the security level of the hidden value of the data from un-trusted servers due to the fact that the DASs need more numbers of k to know the details of the data. If the number of shares decreases to fewer than 3, then it might not be very effective for privacy purposes.

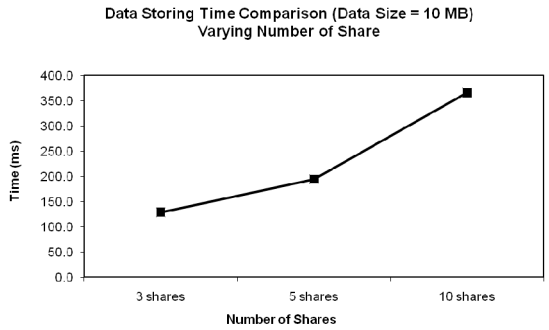

Figure 7. Data Storing Time Comparison, Varying Shares.

Even though the time cost is increased along with the increased number of shares, increasing the number of shares will improve the security level of the hidden value of the data from un-trusted servers due to the fact that the DASs need more numbers of k to know the details of the data. If the number of shares decreases to fewer than 3, then it might not be very effective for privacy purposes.

4.2.2 Data retrieval procedure. For data retrieval, we simulate different types of queries such as the exact match, aggregation and range query. The data retrieval procedure in the NetDB2-MS model starts from rewriting the user's query in the DBMS (n numbers of queries) and then sends these queries, one for each DAS provider, after constructing the polynomial and the order for the secret value. The relevant tuple will be returned to the DBMS to compute the polynomial function on the returned value.

On the other hand, data retrieval in the NetDB2 service focuses on data decryption after the data has been retrieved from the data source.

- **Exact Match Query Process**
  We compare the data retrieval time cost between the secret sharing algorithm in NetDB2-MS and the blowfish algorithm in the NetDB2 service for the exact match query process. Figure 8 shows that the blowfish technique is more costly than the secret sharing algorithm due to decryption costs.

  We also evaluate the outcomes of data retrieval for various numbers of shares in the secret sharing algorithm in NetDB2-MS. Figure 9 shows that data retrieval time increases incrementally with an increased number of shares. On the other hand, we argue that increasing the number of shares will also increase the security level of data because DASs will need to retrieve more values from more shares in order to be able to determine the hidden information in DASs.

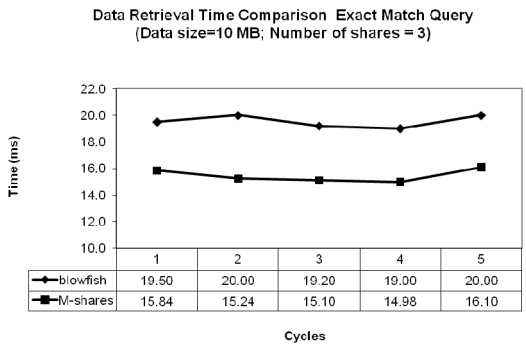

Figure 8. Exact Query Data Retrieval Time Comparison

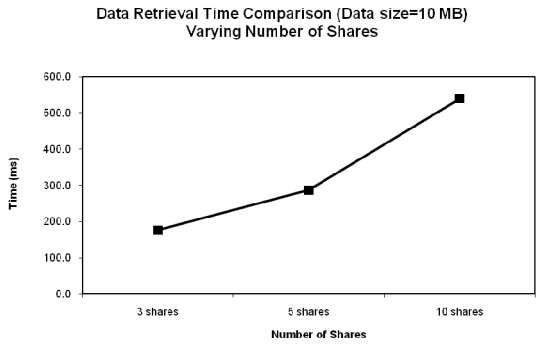

Figure 9. Data Retrieval Time Comparison, Varying Shares.

- **Aggregation Query Process**
  We also compare the data retrieval time in an aggregation query type, where we retrieve the maximum value of the data set. Figure 10 shows that the time cost for the blowfish technique is higher than the time cost for the secret sharing algorithm because the decryption techniques, followed by the retrieval of the maximum values from the data source, take longer than does the retrieval of the maximum values from the shares in DASs in NetDB2-MS.

- **Range Query Process**
  Finally, we compare data retrieval time cost for the range query. In this case, we retrieved the data from the same range value between 1000 and 9000. Figure 11 shows that the decryption techniques and the operation to retrieve different values from the data source in NetDB2 takes longer than retrieving the same values from the shares in NetDB2-MS.

From the simulations, we prove that our proposed solution outperforms the existing NetDB2 practice for both data storing and data retrieval operations. Even though the simulation is not done in the real NetDB2 system, it is a fair indicator of its valuable contribution to the area of improving data privacy in DAAS.

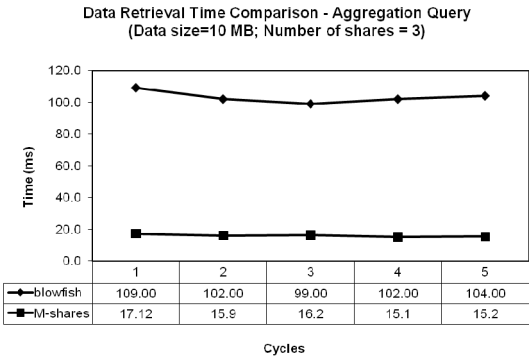

Figure 10. Aggregation Query Data Retrieval Time Comparison

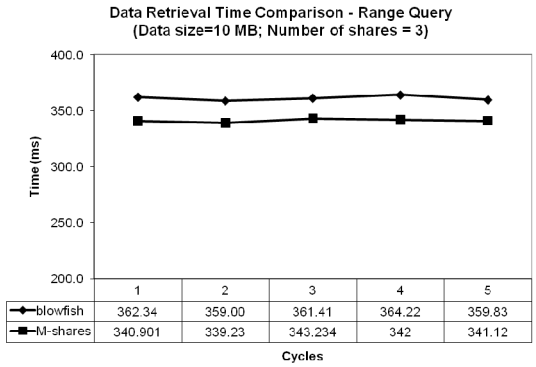

Figure 11. Range Query Data Retrieval Time Comparison

5. Conclusion and future work

The purpose of this work is to ensure data privacy in the DAAS model and to protect valuable information from unsafe or un-trusted servers, especially in users' queries during data retrieval procedure. It was found that the number of solutions for data privacy in DAAS using encryption techniques were higher than those for any other techniques. The use of multi-sharing techniques is considered novel. It is proven to be superior to the encryption technique due to the latter's limitations and its negative impacts such as high time cost for encryption and decryption of the data.

For future work, further evaluation of non-numeric data types and larger data size can be performed. Another area of attention for future work is the idea of substituting simulation for real DAS, such as NetDB2 service, and comparing it with the proposed model. Finally, there are different types of encryption /decryption techniques that we may compare with NetDB2-MS in addition to the Blowfish encryption technique alone. This is evident since different DAASs

that employ encryption, might use several encryption techniques.

References

[1] D. Agrawal, A. El Abbadi, F. Emekci and A. Metwally, *Database Management as a Service: Challenges and Opportunities.* In *ICDE 2009*, pp. 1709-1716.

[2] R. Agrawal, A. Evfimievski and R. Srikant, *Information sharing across private databases*. In *SIGMOD 2003*, pp. 86-97.

[3] R. Agrawal, J. Kiernan, R. Srikant and Y. Xu, *Order preserving encryption for numeric data.* In *SIGMOD 2004* , pp. 563-574.

[4] N. Anciaux, M. Benzine, L. Bouganim, P. Pucheral and D. Shasha, *GhostDB: querying visible and hidden data without leaks.* In *SIGMOD 2007*, pp. 677-688.

[5] S. Aulbach, T. Grust, D. Jacobs, A. Kemper and J. Rittinger, *Multi-tenant databases for software as a service: schema-mapping techniques.* In *SIGMOD* 2008, pp. 1195-1206.

[6] B. Chor, O. Goldreich and E. Kushilevitz, *Private information retrieval*, Google Patents, 1998.

[7] A. Freier, P. Karlton and P. Kocher, *The SSL Protocol, V3. 0*, Available at: http://wp.netscape.com/eng/ssl3/3-spec.htm accessed in August 2009.

[8] Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin, *Protecting data privacy in private information retrieval schemes*. In *STOC 1998*, pp.151-160.

[9] D. Gupta, P. Jalote and G. Barua, *A formal framework for on-line software version change.* In IEEE Transactions on Software Engineering, 22, 1996, pp. 120-131.

[10] H. Hacigumus, B. Iyer and S. Mehrotra, *Providing database as a service*. In *ICDE 2002*, pp. 29.

[11] H. Hacig, B. Iyer, C. Li and S. Mehrotra, *Executing SQL over encrypted data in the database-service-provider model.* In *SIGMOD 2002*, pp. 216-227.

[12] G. Hamilton and R. Cattell, *Jdbc: A java sql api*, Sun Microsystems, 1997.

[13] B. Hore, S. Mehrotra and G. Tsudik, *A privacy-preserving index for range queries.* In *VLDB 2004*, pp. 720-731.

[14] Y. W. Lee, D. M. Strong, B. K. Kahn and R. Y. Wang, *AIMQ: a methodology for information quality assessment*. In Information Management 40, 2002, pp. 133-146.

[15] H .Mei, J. Dawei, L. Guoliang and Z. Yuan, *Supporting Database Applications as a Service.* In *ICDE 2009*, pp. 832-843.

[16] K.-S. Na, D.-K. Baik and P.-K. Kim, *A practical approach for modeling the quality of multimedia data*. In *ACM Multimedia 2001*, pp. 516- 518.

[17] R. Ostrovsky and V. Shoup, *Private information storage (extended abstract)*. In *STOC 1997,* pp. 294 - 303 .

[18] U. Ozan and T. i Gundem, *A Survey on Querying Encrypted XML Documents for Databases as a Service*, In *SIGMOD Record*, 37, 2008, pp. 12-20.

[19] A. Shamir, *How to share a secret.* In *Communications of the ACM*, 22, 1979, pp. 612-613.

[20] R. Sion, *Query execution assurance for outsourced databases*. In *VLDB 2005*, pp. 601-612.

[21] R. Sion, *Secure data outsourcing*. In *VLDB 2007*, pp. 1431-1432.

[22] R. Sion and B. Carbunar, *On the computational practicality of private information retrieval.* In *NDSS,* 2007, pp. 2006-06.