Secure Outsourced Database Architecture

Ahmed M.A. Al thneibat[1], *Bahaa Eldin M. Hasan*[2], *Abd El Fatah .A. Hegazy*[3], *Nermine Hamza*[4]

[1] *Arab Academy for Science &Technology College of Computing and Information Technology, Cairo branch, Egypt.*
[2] *Arab Security Consultants (ASC)*
[3] *Arab Academy for science &Technology College of Computing and Information Technology, Cairo branch, Egypt.*
[4] *Institute of Statistical Studies and Research (ISSR),Computer Science and Information Department, Cairo University.*

Abstract

With fast increasing of database outsource and its gained benefits such as increase data availability, reduce the cost and more…, we present at this paper a new model that split the architecture of database outsource into three components, database owner/clients, Secure Meta Mediator SMM and Database Service Provider. The model architecture aims to improve in efficient way the components of secure database outsource include; Query Correctness, Data Confidentiality and User Data Privacy**.** The model architecture makes the most of data processing performed through the SMM, reduces computation and communication overhead by partially encrypt the data and information and increase data confidentiality through using deferent encryption techniques.

Keywords:
SMM, DSP, DAS, Semi-Encrypted

1. Introduction

Today's database outsource has become one from the interested trends for the organizations. In such an outsource database “Database-as-Service” in DAS model [16], clients outsource data management to a “Database Service Provider” that provides online access mechanisms for querying and managing the hosted data sets. Database outsource includes number of benefits such as; Decrease the cost 5-10 times from initial equation, Increase database availability, Offer much cheaper services and offer expertise consolidations [1, 16]. However it’s also including number of critical issues such that related of the database service provider, it couldn’t be a trusted party which mean it’s not allow to read, write and modify the organization data and information. For successful secure database outsource, the organizations should take in consideration the security issues mentioned above is covered, the coverage of these security issues accomplished by the implementation of three security components; Query Correctness, Data Confidentiality and User Data Privacy [3]. The most of approaches of successful implementation of database outsourcing introduces number of logarithmic and hardware solutions that interests in how to cover these security aspects. This paper organized as follows. In section 2, we present the previous works of each of database outsource security components. In section 3, we present our architecture model and its components. In section 4, it is about database metadata and query construction management based on our architecture model. In section 5, we present our architecture security components. In section 6, we project our model Architecture Construction and functionality. In section 7, it is the conclusion of our proposed secure outsourced architecture.

2. Previous Works

The most of previous works that related of secure database outsource cover partially one of its security components by uses a logarithmic or hardware solutions. In more details, for the issue of query correctness which briefly defined as technique or mechanism that makes the client able to verify the integrity and completeness of the query result [1]. There number of solutions proposed to cover this security component such as; *Public-Key Digital Signature* Schemes that combination between asymmetric encryption and

Manuscript received May 5, 2010
Manuscript revised May 20, 2010

collision-free hash function that used for authentication and integrity of signed messages among the parties [2]. *Merkle Hash* Tree is also one from the previous solutions that aim to authenticate the result of simple range queries in a publishing scenario, which the data owners delegate the role of satisfying user queries to untrusted publisher [4]. There is number of more advanced and modern solutions has been proposed also to cover the security aspects of query correctness. *The Efficient Two-Party Authentication Protocol* it based on cryptographic hashing, the database owner outsource its data structure and verifies that the answers of the queries are valid, by using a skip list, which is known efficient randomized realization of a dictionary, and the client store only a single hash value [5]. *Key Nearest Neighbor (KNN)*, the mechanism used to verify that their answer for k nearest neighbors on a multidimensional dataset is complete to satisfy the requirements of the query correctness. The proposed mechanism based on signature chain concept, it verifies the *KNN* answers are complete [6]. In general, the drawbacks of these solutions it highly increase the computation and communication overhead more than our selected solution. It is not include framework drawing, which shows how the solution performs the operations and steps among the parties. Did not illustrate how it could be integrate with other database outsource security components and there is no clear vision about how to deal with multi owner databases and about the complex queries operations.

For the issues related of the data confidentiality, the proposed previous solutions aim to encrypt the organization data and information by using strong encryption algorithm that decreases the probability of data and information compromise during the transition of data and information between the client and server, even if the server is malicious [1]. The solution of *Shared and sharable encrypted data for untrusted servers*, its multi-users searchable data encryption, each user will be able to encrypt and decrypt the inserted data by other users without need to know the users keys [7]. Another solution it called *Distributed Protection Mechanism*, which characterized by multiple data owners sharing data with large members of intermediaries and final recipients [8]. Another solution it called *Dynamic Group Key Management*, it constraint in how to manage the distribution of users keys and reduce the processes of keys expiration or revocation by constructing a group of users and assign members into the groups and also revoke the users from the group. The goal is reduce the computation cost that will only affect the group only not the whole database [9]. In general, the drawbacks of these previously proposed solutions relates of its complexity in performing the operations. Its encryption and decryption time costs. It depends of encryption algorithms. It does not show how it could be working with huge systems especially with multi-owner systems. There is no covering of the indexing scheme and the management of key's distribution and revocation. It is not applicable to be integrating with other database outsource security components and does not include a solution provable experiment illustrates its applicability in real life model.

The third security component of secure database outsource is User Data Privacy, which aims to; the malicious server shouldn't be able to analyze and learn anything about the client's query patterns by performing such as query statistical attacks [1]. There number of previous proposed logarithmic and hardware solutions that aims to cover these components. The Usable PIR is one from the proposed solutions that integrates and enhances number of logarithmic solutions and protocols (PIR, ORAM and strawman) with hardware solution IBM Secure Co-processor SCPU, the solution aims to guarantee access pattern privacy against a computationally bounded adversary, in outsource data storage [10]. There is another solution called the *technique of Oblivious Search and Updates on Outsourced Search Trees*, it focus in allow the clients to operate on their outsourced tree-structured data on untrusted servers without revealing information about query result and the outsourced data itself [11]. Another solution cover the component of user data privacy in deferent way, it is the technique of *privacy-preserving indices*, its focus in filtering out tuples of the query predicates that involve sensitive attributes. The proposed technique involves partitioning each attribute domain into a finite number of regions in an equal-depth or equal-width manner and assigning each region a unique random tag (bucket-id) and aim to balance between the performance and the privacy [12]. In general, the drawbacks of these previously proposed solutions relates of its dependability of hardware solutions. Its complex operation that effects the communication and computations cost. It requires additional encryption processes that may increase the communication and computation processes. Do not include a solution provable experiment illustrates its applicability in real life model and does not illustrate how it could be integrate with other security component (Query Correctness and Data Confidentiality).

3. Model Architecture

3.1 Overview

Our model based on semi-encrypted database outsource, which means there is certain type of data that related of database owner and it assume confidential data and the other is not. It is also, there is another data type that related of database clients and it assumed confidential data. each of this parties own his keys for encryption and decryption, the master keys able to be stored by the SMM or database owner smart tokens. On the other hand, the outsourcing data will encrypt by using secret keys stored in outsourcing database side. Another type of encryption happen over the metadata structure which use SMM stored keys. This model achieves data confidentiality, user privacy and query correctness. The proposed architecture as shown below:

The database tables contain data specified for the organization (Database owner) and clients. Therefore, the encryption for the database includes metadata, relations, tables, fields and tuples will be separately independent. The client's data will encrypted by using the client keys. These keys can be stored on SMM server or in smart tokens according to the client preferences, on the other hand other outsourced data will encrypted by using a secret keys stored in outsourced database owner side. Another type of encryption happen over the metadata structure which use SMM stored keys. Our model aims to achieve data confidentiality, user privacy and query correctness.

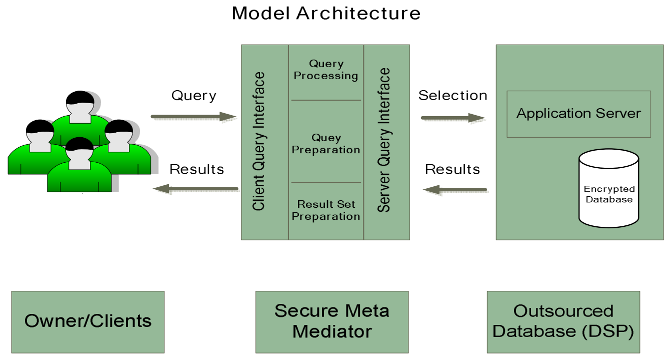

Fig. 1 Outsource database model based on Secure Meta Mediator

3.2 Architecture Components

Let us start to define the components of our secure outsourced database architecture:

Database Owner: An organization started to outsource its own database to be hosted external at DSP (Database Service Provider) side. It is at the first initiation encrypt database structure *by using keys store at (SMM) and smart tokens.* The database owner categorizes the privileges and assigns these with keys that allow the organizations deferent level employees and clients to gain access to the data. The database owner who is only one knows the structure of database and store the Metadata in the SMM server.

Database Clients: Database clients are actually the data users, who read, write and modify the database records according to their privileges. The clients may database-authorized employees in organization or a client such as bank customers. These clients encrypt/decrypt the data by using a secrete key stored on SMM server and may store in smart tokens (optional).

Secure Meta Mediator SMM: Is the middle server owned by the organization (Database Owner), it contains an encrypted copy of the Metadata of the database and keys. It is available to the clients according to their authorization level. The SMM include three core components, Client Query Interface, Query Processing and Server Query Interface. Let us start to define the components of SMM:

- **Client Query Interface:** It is responsible for authenticate clients through secure authentication protocol such as Secure Socket layer (SSL) and authorize the clients according to the client's assigned privileges through such as Database Access Control Lists (ACL).
- **Query Processing:** Query Processing consists of two modules (Query Preparation and Result Set Preparation), the first module process the query (parse, and fetch) by using its own Metadata. The second module prepares the result set of data and information to be transmitting to the client/owner.
- **Server Query Interface:** It is responsible for initiate encrypted connection for execute (send/receive) data selection to/from the outsourced database.

Outsourced Database: Is an encrypted data that stored at the DSP servers, it help the organization to gain the number of benefits such as high availability, cost reduction and more..., Database outsource must be unable to be decrypted, unable to be analyzed and unable to be tempered by adversaries through using number of crypt-analysis solutions and techniques

4. Database Metadata and Query Construction Management

In traditional models, DBMS (Data Base Management System) contains a special component called Query Processor; its task is to care of arranging the underlying access routines to satisfy a given query. The relation queries processor is consist of four processes; Query parsing, Query Rewrite, Query Optimizer and Plan Executor [13]. Fig 2 illustrates the traditional model for query processing.

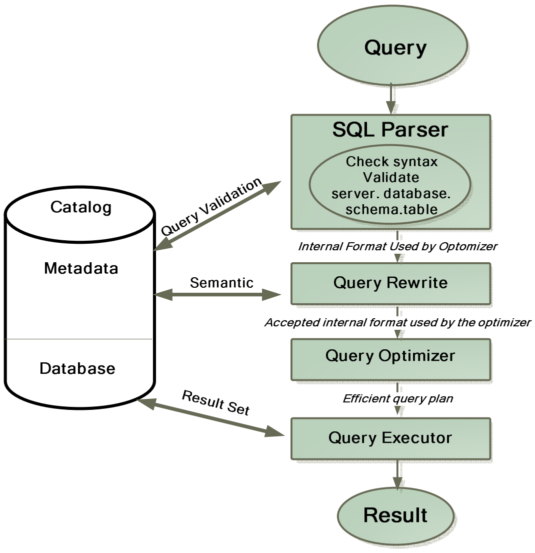

Fig. 2 Traditional DBMS Query Processing

Based on Metadata Management in DAS scenario [17], at the previous design, the query processing and the metadata are stored and processed at the client/owner side as shown Fig 3.

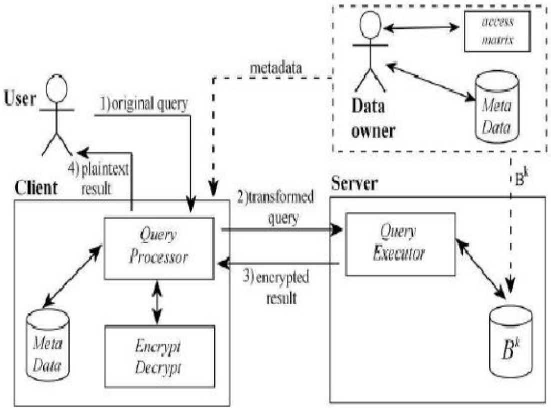

Fig. 3 DAS Scenario [17]

In more advance details, the metadata is consisting of three subtypes, Authorization Metadata, Descriptive Metadata and key Management Metadata. The Authorization Metadata include information about the access control policy defined by the data owner, it is very sensitive therefore it will stored at the SMM server. The Descriptive Metadata, it is simply the data description; describe the structure of the encrypted database and its similar to the system catalog that include (tables relations, tables indexes tables, table methods and encryption algorithms) its will stored at the SMM. The key management metadata include information about the key derivation method and information about the portion of the user tree hierarchy associated with the corresponding user. Such a sub-hierarchy allows a user to derive the keys necessary for decrypting the data for which she has a read privilege, in our model in section 5.2 we illustrate how it has managed and distributed [17].

In our model at the initiation phase, the database owner imports the structure of database (Metadata) on the SMM to switch number of tasks from database service provider to perform at the SMM. The main advantage of this task to make the SMM be able to perform number of queries processing tasks (Parsing and optimizer), then the execution phase will be at the outsourced database.

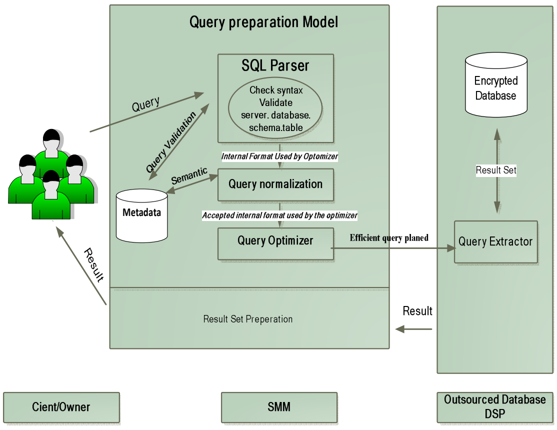

Fig. 4 SMM Query Processing

5. Architecture Security Components

For successful secure database outsourcing, there number of security components should be included. Let us start to illustrate how these components covered in our architecture model.

5.1 Query correctness

As mention before in section 2, Query Correctness is a technique or mechanism makes the client able to verify the integrity and completeness of the query result. Query correctness aims to insure the outsourced data could not temper by the untrusted server (Database Service Provider), and to insure the query executed successfully and the result does not truncated by the server [1].

We build the component of Query Correctness in our model architecture based on *Query Execution Prove* mechanism [3], for query correctness for database outsource framework. The solution handles arbitrary query types and features reasonable overhead factors. The solution framework based on two parties the database owner and database service provider. The solution includes, number of advantages such as, it uses hashing algorithm MD5 which one form the fastest hashing algorithms. It accommodates our architecture of database outsource and help to minimize the communication and execution overhead. It includes the most of required aspects for applicable real life solution (query processing, data updates, hashing algorithm and experiment). The proposed solution proves success probability of cheating less than 5% with an execution time overhead fewer than 25%.

We accommodate and integrate the solution of *Query Execution Prove* to satisfy our architecture model. The solution parties at our model are the Secure Meta Mediator SMM and the outsourced database. The solution enhancements at our model focus in switching the data segmentation and computational overhead from the client's side to the Secure Meta Mediator. The main advantage of these enhancements is to perform all the operations that related to query correctness at the Secure Meta Mediator.

On our proposed Solution, the SMM send a batch of queries every time concatenated with a challenge token. The challenge token is C $(\mathbb{Q},x,\epsilon) = \{H(\epsilon\|\rho(Qx)),\epsilon\}$ is sent together with the batch of queries. On return, query batch execution is proved if X'=X holds. In the first phase, the challenge token used for every batch of queries $\mathbb{Q}$ ={Q1,…,Qb} to be executed at outsourced database. The SMM picks a secret number X ∈ [1,b] and a unique one-time use random Nonce ϵ and computes a query batch challenge token, C($\mathbb{Q}$,x,ϵ) = {H(ϵ||ρ(Qx)),ϵ}, where "||" denotes concatenation [3].

In other words, SMM executes the corresponding query (Qx) over the outsourced database and computes a one-way cryptographic hash of the nonce concatenated with the query result. This token then used in the next step to challenge the service provider to prove actual query execution. The SMM compute the challenge tokens by periodically retrieve a generate set of fake queries and compute and store the associated challenge tokens (For later use). This has the advantage of reducing communication costs.

The second phase is when the SMM starts to assign a query by submitting a batch $\mathbb{Q}$={Q1,…,Qb} with challenge token C($\mathbb{Q}$,x,ϵ). The service provider then executes the queries in the batch and computes the value for x, by identifying which of the queries maps to the received challenge token (it can do so by computing the associated challenge tokens for each executed query, for the given nonce ϵ). Let this value be X'. This will constitute the main query execution proof.

The service provider then at the third phase returns both the query execution proof X' and the query results for the batch ρ($\mathbb{Q}$)={ ρ(Q1),…,(Qb)}. The SMM verifies that indeed (X' = X); if true, it provides a statistical proof that the queries were performed correctly over their target data sets and then transmit the results to the clients.

5.2 Data Confidentiality

The data and information must be encrypted by using strong encryption algorithm that decreases the probability of data and information to be compromised during the transition of data and information between the client and server (un-trusted server), even if the server is malicious. In other words, Data confidentiality is the protection of private information from sniffing or leaks when it is stored, or transmitted across vulnerable networks such as the Internet. Actually, the confidentiality of data varies, depending on organization types and data classifications [1].

We build the component of data confidentiality on our model architecture based on Mixed Cryptography Database solution [14]; it includes number of advantages that accommodate our model architecture. The solution based on semi-encrypted database model. Our model data classification, consists of three types, it listed below:

1. Metadata: its data about the data includes relations, fields name and more...
2. Owner Data: includes organization data.
3. Client Data: client is such as employees and customer's private data.

The encryption solution in our model encrypts the three types of data with three types of keys. We used a symmetric algorithms and symmetric keys (e.g. 3DES, AES…etc.). The solution includes, number of advantages, accommodates our model architecture such as; it is very simple and applicable to linked with other database outsource security components (Query Correctness and User Data Privacy). The solution based on semi-encrypted database, which mean there types of data classified as confidential will encrypted and the other is plaintext. It is applicable to apply more than one encryption algorithms such as one for the clients and the second for the database owner.

The first data classification is for database Metadata. At the initiation steps, (designing and creation), which has been done by database owners and they are the only authorized to change or modify it. This authorized process such as create, modify and delete in the structure of metadata controlled by database owner keys stored in smart tokens or in SMM server according to database owner preferences. The relations, table name and fields they create, is stored encrypted in the SMM. The keys use in this encryption/decryption is stored in the SMM server to be use in query processing when needed.

The second data classification is for Owner Data: it is simply the organization data such as employees, financial and customer's accounts, some of these encrypted and the other in plaintext. Keys used for encryption/decryption stored in SMM server.

The third data classification is for Client Data: it is simply data owned by employees or customers. In our model, its enable the client to store the data and information as encrypted or in plaintext using symmetric key stored either in SMM, smart token or both. The following tables illustrate our model data classification based on semi-encrypted mechanism.

Table1: Relation Name table as a plaintext

Field1	Field2	Field3	Field4
Client data	**owner data**	Client data	**owner data**

Table2: E(Relation Name)$_{\text{creation key}}$ Is table 1fter encryption

E(Field1)$_{\text{creation key}}$	**E(Field2)**$_{\text{creation key}}$	**E(Field3)**$_{\text{creation key}}$	**E(Field4)**$_{\text{creation key}}$
E(Client data)$_{\text{client key}}$	E(owner data)$_{\text{owner key}}$	E(Client data)$_{\text{client key}}$	Owner data (Not encrypted)

As shown above Fig 5, table2 is uses three deferent types of keys related of our model data classification: creation keys (Metadata), owner keys (owner data) and client's keys (client's data). The following table Fig 6 illustrates keys classification.

Table 3: Keys Classification

Key name	Key owner	Storage place
Creation key	Database owner	SMM
Owner key	Database owner	SMM
Client key	Database Client	SMM and/or smart token

There exists another type of keys not used in query processing or encrypted stored tables. These keys owned by database owner, stored in smart tokens or in SMM server, it is used only when any modification of metadata. About Keys lifetime, it is a specific period, during which a cryptographic key setting remains in effect. These keys must be expiring and regenerate based on the organization policy. Of course, the database will be altered and re-encrypted data.

Changing owner data by:

For every value desired to be changed

UPDATE (relation-name) **SET**(field-name) = $E_{\text{new SMM key}}(D_{\text{SMMkey}}(\text{oldvalue}))$ **WHERE** some-condition

Changing the Client data by:

For every value desired to be changed

UPDATE (relation-name) **SET** (field-name) = $E_{\text{new client key}}(D_{\text{client key}}(\text{oldvalue}))$ **WHERE** some-condition

Changing the metadata for outsourcing database:

For every field in the desired relation

ALTER TABLE (relation-name) **CHANGE COLUMN** (old-field-name) $E_{\text{new creation key}}(D_{\text{creation key}}(\text{old-field-name}))$ (old-field-name TYPE)

For every desired relation

ALTER TABLE (old-relation-name**)** **RENAME TO** $E_{\text{new creation key}}(D_{\text{creation key}}(\text{old-relation-name}))$

Let us start to illustrate it in simple example the database table EMP before, and after encryption by using deferent types of keys:

Table 4: EMP table before encryption

Emp_ID	Emp_name	Dept_no
23	**Layla**	**12**
42	**Ahmed**	**7**
15	**Asmaa**	**6**

Table 5: Tmn=E(EMP) EMP table after encryption

Yghre	Uiuiuy	dfgdg
Gh	**Eiuy**	**Tr**
Jk	**Uiy68**	**9d**
Df	**Uf74**	**6d**

E (Dept_no)$_{\text{MMS}}$

E(Emp_mane)$_{\text{Client}}$

5.3 User Data Privacy

As mentioned before, data access privacy is simply at the database outsourcing implementation, is the malicious server shouldn't be able to analyze and learn anything about the client's query patterns by performing such as query statistical attacks [1].

We build the component of User Data Privacy on our model architecture based on Blind Custodian solution. The solution offers information dissociation, which means the server stores only fragments of data and information that considered safe and not violate the privacy [15]. The solution accommodates our architecture model and includes number of beneficial advantages that satisfy our architecture model, the advantages of the Blind Custodian solution such as; it is not rely on encryption. Applicability to be operates with complex queries and multi owner systems. Applicable to be integrate with other database outsource security components.

The database owner, at the preparation phase before outsource the data in an initial steps, the database is decomposed into fragments $\mathrm{F1}, \dots, \mathrm{Fk}$, these fragments is indexed (Ii),Fi $(\mathrm{i} = 1, \dots, \mathrm{k})$ then it ciphered $\mathrm{C} = (\mathrm{I1}, \dots, \mathrm{Ik})$ these ciphers is stored at the client/owner sides. Let us start to illustrate the basic construction of the solution query processing based on our architecture model, the SMM submit a query Q on $R\ database\ relation,$ Q is transformed to a query Q^* (encrypted Query) to the server database F_1,…, F_k. The evaluation of query on the server's database returned A^* to the SMM, the SMM then transforms this result to a new relation using its cipher C (this result is denoted $T^{-1}(A^*)$ the enviers of A^*). To this, the SMM applies final processing (Q') to obtain the answer A and then send it to the client.

We choice *Frugal Join* technique, it is one of two presented techniques by the solution author, after the SMM receives the query from the client and send it to the outsourced sever. It is simple at the First the server performs selections and projections on its fragments and sends the results to the SMM. The SMM then joins the results uses its cipher and applies the final selection and projection, and then send it to the client. In practical example, the SMM send a query that received from connected clients to the Database Service Provider (Outsourced Database), about the id's of female employees who earn over \$80,000 and have been in employment more than half their lives:

Q :
select Eid
from Employee
where Salary > 80,000 **and** Gender = 'female' **and** (2005 − YearHired) > 0:5 ∗ (2005 − YearBorn)

The server will split the query into two queries as shown below:

Q_1^*:
Select I_1, Eid, YearHired
From F_1
Where salar> 80,000

Q_2^*:
Select I_2 , YearBorn
from F_2
where gender = 'female'

The server then sends the result of two queries to the SMM, who then concludes the processing with a query, and joins the answers through its cipher and then extracts the final tuples that constitute the answer A and send it to the client/owner:

Q':
Select Eid
From A_1^*, A_2^*, c
Where $A_1^*.\,I_1$= c. I_1 **and** $A_2^*.\,I_2$=c. I_2
And (2005 − $YearHired$) > 0.5∗(2005 – $YearBorn$)

Finally, the SMM sends it to the client. Note the *Frugal Join* technique will perform huge number of process however, it also increase the level of confidentiality [15].

Model Architecture Construction and functionality

Now, let start to illustrate our proposed model architecture construction and functionality in advance details:

6.1 Database Preparation

At the preparation phase, the database owner starts to create a database schema and encrypt it, the datasets will stored in the server (Database Service Provider) side and the metadata will be stored at the SMM. The encryption will be at the relation level, attribute level, tuple level, and element. For the relation level and attribute level its will encrypt/decrypt through the owner creation key and about the tuple level and element level it will be encrypt/decrypt through owner/clients keys, these keys is stored at the SMM for the owner side and about the clients keys, it's may store in the SMM server or in smart tokens. About the initiation, it described it details in section 5.3 and about the keys it described in details in section 5.2 it illustrated in Fig 5, 6 and 7.

6.2 SMM Functionality

As mentioned in section 3.2, the clients will connect through secure connection such as SSL/IPSEC to the SMM clients query interface. After authentication process is complete, the client then will submit query, The queries will be processed through the SMM query processing steps, shown Fig 5, and illustrated in advance with steps in section 4, the then the query will transmitted through the SMM server query interface to the outsourced database. It's necessary to mention here the processed and transmitted query is secured by the applied query correctness solution section 5.1, and user data privacy 5.3 and the data is already encrypted section 5.1, that what we aim for. Finally, the database server will send the results to the SMM server query interface and its will checked it section 5.1, it will be prepared section 4 by SMM query processing and then sent to the client through the SMM client query interface.

6. Conclusion

In this paper, we presented a new architecture of semi-encrypted outsourced database. The architecture model splits the previous DAS Model [16] design from two main parties (owner and outsourced database), into three (client/owner, Secure Meta Mediator SMM and outsourced database). The main goal of the design of the architecture is to redistribute the functionality of outsourced database by splitting the operations between outsourced Database and the trusted server (SMM). The model implements Database outsource concepts and increase the security level by performing the most of query processing tasks at the trusted SMM. This model supported the three main security components in the outsourced database (query correctness, data confidentiality and user data privacy) and integrate it with number of enhanced and modified selected solution for each of these security components.

References

[1.] Radu Sion (2008)" Towards Secure Data Outsourcing" in Michael Gertz, Sushil Jajodia (2008) "Hand Book of database Security application and trends" Springer Science Business Media, LLC.

[2.] Viet Hung Nguyen, Tran Khanh Dang (2008)" A Novel Solution to Query Assurance Verification for Dynamic Outsourced XML Databases "in JOURNAL OF SOFTWARE, VOL. 3, NO. 4, APRIL 2008.

[3.] Radu Sion (2005) "Query Execution Assurance for Outsourced Databases" in the 31st VLDB Conference, Trondheim, Norway.

[4.] Feifei Li, Marios Hadjieleftheriou, George Kollios, Leonid Reyzin (2006) "Dynamic Authenticated Index Structures for Outsourced Databases". In SIGMOD 2006, June 27–29, 2006.

[5.] Charalampos Papamanthou, Roberto Tamassia (2007) "time and space efficient algorithm for tow-party authenticated data structures" in Sihan Qing, Hideki Imai, Guilin Wang (2007) "Information and communications security" Springer.

[6.] Weiwei Change, Kian-lee Tan (2007) "Authentication KNN Query Results in Data Publishing" In Willem Jonker and Milan Perkivic(Eds.) (2007) "Secure Data Managements" Springer.

[7.] Changyu Dong, Giovanni Russello, Naranker Dulay (2008) "Shared and Searchable Encrypted Data for Untrusted Servers" in Vijay Atluri (2008) "Data and Applications Security" 22 edition, Springer.

[8.] Gerome Miklau (2005) "Confidentiality and Integrity in Distributed Data Exchange".

[9.] Alla Lanovenko and Huiping Guo (2007) "Dynamic Group Key Management in Outsourced Database" Proceedings of the World Congress on Engineering and Computer Science 2007.

[10.] Peter Williams and Radu Sion (2008) "Usable PIR" in proceeding of NDSS (2008).

[11.] DANG Tran Khanh (2005) "Oblivious Search and Update for Outsourced Tree-Structure Data on Untrusted Servers" International Journal of Computer Science & Applications *Vol. II, No. II, pp. 67 – 84.*

[12.] Bijit Hore, Sharad Mehrotra, Gene Tsudik (2004) "A Privacy-Preserving Index for Range Queries" Proceedings of the 30th VLDB Conference 2004.

[13.] B Joseph M. Hellerstein , Michael Stonebraker and James Hamilton (2007) "Architecture of a Database System" Foundations and TrendsR in Databases Vol.1, No. 2 (2007) 141–259 c 2007 J. M. Hellerstein, M. Stonebraker and J. Hamilton DOI: 10.1561/1900000002.

[14.] Hassan Kadham, Toshiyuki Amagasa, Hiroyuki Kitagawa (2009) "A Novel Framework for Database Security based on Mixed Cryptography" IEEE Xplore.

[15.] Amihai Motro and Francesco Parisi- Presicce (2005) "Blind Custodians: Database Service Architecture that Supports Privacy without Encryption".

[16.] H. Hacigumus, B. R. Iyer, and S. Mehrotra " (2002) Providing database as a service" In IEEE International Conference on Data Engineering (ICDE), 2002.

[17.] E. Damiani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati (2005) "Metadata Management in Outsourced Encrypted Databases" Springer -Verlag.

Ahmed M.A. Al thneibat received the B.S in Computer Information System from The Higher Inset. for specialized technological studies, Egypt, 2001. During 2001-2005, he worked in Civil Status and Passport Department (CSPD), Ministry of Interior of Jordan, as Programmer and Database Administrator. Since 2005 until now, working in Arab Administrative Development Organization (ARADO) – Arab league, as Oracle Database Instructor and Information Security Specialist. During working experience, he received number of certifications from Microsoft, Oracle, fortinet and EC-Council.

Bahaa Eldin M. Hasan received the B.Sc. and M.Sc. degrees, from faculty of Engineering (Shoubra), Zagazig University in 1978 and 1987, respectively. He received the Dr. Eng. degree from Ain Shams University under supervision of Tokyo institute of Technology in 1994. Bahaa has served the National Defense Council Service for 26 years. During his 26 years, he was engaged in general National Defense Council duties. He awarded the Order of Merit- Second grade from the president of Egypt. Bahaa left the National Defense Council in 2006 and went to work for his own privet business "Arab Security Consultants (ASC)".Bahaa is an expert specializing in such areas as: Data security, network security, computer security, Ethical hacking and countermeasures, and Smart card and smart token applications for securing the data and information. Bahaa is still involved in the training of security officers as well as for security staff for several Arab world organizations.

Abdelfatah A. Hegazy received the B.E. degrees, from the Military Technical Collage, Cairo, Egypt, 1978. In 1982 he received the M.Sc. In Computer Sciences from George Washington University, USA. Dr. Hegazy received the Ph.D. Degree Computer Sciences from George Washington University, USA, in 1985. After working as an assistant professor (from 1985) in the Dept. of computer enginering operation research, the Military Technical Collage., and an associate professor (from 1990), he has been a professor at College of Engineering at the Arab Academy for Science and technology. Since 1998. His research interest includes: Information Systems Planning; E-Commerce, E-Government, Information Systems Security, network security ,knowledge Management, Web Intelligent Systems and Enterprise Resource Planning Systems. He is a member of IEEE, ACM, AIS, AANIS, and CSS-Computer Scientific Society Egypt.

Nermin hamza is Assistant Lecture in computer and information sciences department, the Iinstitute of Statistical Studies and Research (ISSR), Cairo University. Nermin got B.Sc. degree in computer science in (2000) in and M.Sc. degree in computer science in (2005) from faculty of computer and information, Cairo University. She works in Data Security and Database area. She taught some basic courses in computer science. Nermin published in the International Conference on Informatics and System INFOS2004, and scientific conference on cyber crime& information security, ACU, 2009. Nermin hamza is now studying a PhD in the field of information security and works in working groups under the supervision of "Arab Security Consultants (ASC)" with special interest in the application of smart token.