# ACCESSING SECURED DATA IN CLOUD COMPUTING ENVIRONMENT

Hasan Omar Al-Sakran

Department of Management Information Systems, King Saud University, Riyadh, Saudi Arabia

***ABSTRACT***

*Number of businesses using cloud computing has increased dramatically over the last few years due to the attractive features such as scalability, flexibility, fast start-up and low costs. Services provided over the web are ranging from using provider's software and hardware to managing security and other issues. Some of the biggest challenges at this point are providing privacy and data security to subscribers of public cloud servers. An efficient encryption technique presented in this paper can be used for secure access to and storage of data on public cloud server, moving and searching encrypted data through communication channels while protecting data confidentiality. This method ensures data protection against both external and internal intruders. Data can be decrypted only with the provided by the data owner key, while public cloud server is unable to read encrypted data or queries. Answering a query does not depend on it size and done in a constant time. Data access is managed by the data owner. The proposed schema allows unauthorized modifications detection.*

***KEYWORDS***

*Data Outsourcing; Privacy; Public Cloud Severs; Access Control; Hash Function.*

## 1.INTRODUCTION

Recent advances in IT technologies and development of high-speed communication infrastructures resulted in cloud computing becoming an excellent alternative to self-hosted data storage. Hardware and software resources (infrastructure, storage, applications, etc.) are provided as cloud computing services that can be accessed remotely [1]. Essential characteristics of cloud computing such as rapid elasticity, measured services, on-demand self-service, ubiquitous network access, and resource pooling have been described by Yang et al. [2]
Services provided by cloud computing:

- IaaS (Infrastructure as a Service) - storage and networking services or virtual servers to subscribers;
- PaaS ( Platform as a Service) - an environment for development of new applications that run on the provider's infrastructures and can be accessed by the subscriber's customers via the Internet;
- SaaS (Software as a Service) - software running on the provider's infrastructure;
- MSP (Managed Service Providers) - specialized services enhancing existing IT services like desktop management services, security management services, e-mail anti-virus, anti-spam services.

Using cloud database servers for storage of an organizational data has its obvious advantages such as access from any device (stationary and mobile) and any location, consolidated expertise of cloud computing providers, high speed query processing, lower initial acquisition, operational and maintenance costs, ability to store huge amounts of data, high availability, etc. All this allows organizations to concentrate on doing their business instead of thinking about its technical side.

Four deployment models of cloud infrastructures have been defined as public, private, community and hybrid. When the infrastructure is under the owner's control, meaning it is owned and managed by the customer and located on customer's premises, it's known as a private cloud. If the infrastructure is under the cloud service provider, meaning it is located off premises and under control of and managed by a cloud service provider, it is a public cloud. In a public cloud customer does not have control over his data, and access to this data can be granted inadvertently to untrusted parties. Several organizations may share infrastructure and support a community that has shared concerns. It may be located on or off premises and managed by a third party or the organizations themselves and known as community cloud. Any combination of two or all three clouds is called hybrid.

The main obstacle to the wide acceptance of cloud storage, even with all advantages of public cloud infrastructure, is concern over the data confidentiality and integrity and possibility of access by untrusted users. Only when cloud providers can provide data security guarantees, the customer will be assured of data safety on the cloud from internal and external threats. The above concerns are addressed in this work by applying a data-hosting model where networks, storage and other devices are shared by several organizations while protecting the confidentiality, integrity, availability and privacy of customer's data achieved based on strong cryptographic guarantees.

Organizations must encrypt the data preceding storage of their data on a public cloud server to guarantee confidentiality. The proposed data encryption schema is using two encryption keys: one for the owner organization and the other for the cloud provider. Two keys must be used to access the data. Only the owner can update data. The service provider cannot view the database content, thus guarantee read-and-write access pattern privacy.

Data owner, clients and cloud service provider communicate over channels safeguarded by keys exchange because the client devices (smart phones, PDAs, etc.) may have inadequate processing capabilities and cannot perform computationally expensive operations, for example asymmetric encryption of blocks of data. Data owner provides clients with an encrypted index to access data.

Searching encrypted data is not a simple task. Usually a client may need to retrieve all data from the cloud server to the local device, decrypt it, and only then the search can be performed. To enable searching over the data, the customer has to store an index locally, or download all the (encrypted) data, decrypt it and search locally. The first approach obviously negates the benefits of cloud storage (since indexes can grow large) while the second scales poorly. Time, effort and bandwidth are wasted. For a mobile user this solution is not acceptable due to short battery life and limited capabilities of his device. This study proposes to transfer all necessary processing from a client device to a cloud server.

The aim of this research paper is to develop an encryption mechanism that will enable users to search encrypted data in a public cloud computing environment. This research is based on author's previous theoretical work [3]. The proposed application can be used to search encrypted databases without compromising security of data and participating parties' privacy in a public cloud environment. Several hash functions can be considered for the generation of encryption

keys. Complicated techniques based on logical operations or the multi-round iteration of some available ciphers can be used for realization of these functions.

The proposed cryptographic system applies collision-free one-way chaotic hash functions [4] to manage keys. These functions possess attractive for cryptography characteristics, namely unpredictable random-look nature, determinism and sensitivity to the initial value. The hash code improves the security of the information systems by permitting an arbitrary identifier to be replaced by a standard fixed-length code. This code is used as an index for referencing related data in response to a user's query. Index creation by the data owner will make search in cloud computing easier and faster. In addition, this method simplifies the system's structure, increases search efficiency, and reduces the size of data and communication delays.

The rest of the paper is organized as follows. Section 2 briefly reviews related work for search on encrypted outsourced data. Section 3 presents details of the proposed architecture including access procedure. Section 4 discusses security analysis of the proposed method. Section 5 concludes the paper and suggests direction for future work.

## 2. LITERATURE REVIEW

A number of systems and methods have been developed to resolve data security and access control issues in cloud computing [5-10]. Dang [11] gave a list of the most important security issues of the cloud computing: data confidentiality, user and data privacy, query assurance, secure and efficient storage.

Searching encrypted data in a cloud is one of the issues that cause concern. Song et al. [12] have been one of the first to study this problem. The authors used a symmetric encryption method for word-by-word encryption of text and storage the resulting blocks as a file. The same technique must be used by a client for encrypting his query, which was sent to the cloud server, and a sequential block-by-block search was conducted. The entire database must be accessed in such a case. This time consuming procedure is not practical for searching on large data. A number of researchers devised secure index methods with keyword indexes saved on the server [13-15]. The index can be used for searching when required. Brinkman et al. [16] developed an algorithm for searching databases in XML format. Goh [17] used a trapdoor generated by a secret key in his efficient secure searching technique over encrypted data and developed a secure indexing model. His method allows to check if a particular word is present in searched data in a single operation and provides semantic security mechanism against an adaptive chosen keyword attack. Using the trapdoor is the only way to find anything from the index, but this method does not support multiple keywords search.

Boneh et al. [18] devised a searchable public key encryption schema based on a sequential search at the server. Authors developed two techniques; one is based on bilinear maps, and another on trapdoor permutations. Dai et al [19, 20] developed a PKI-based encryption technique that allowed clients direct access to cloud data and proposed dynamic management of distributed resources and access to them by remote actors. Kan Yang et al. [21] designed secure data access control mechanism for multiple authority cloud storage and revocation method which provides forward as well as backward security. A decentralized access control technique to support anonymous authentication was proposed by Ruj et al. [22]. Bamiah et al. [23] developed on fly encryption of data at storage server, and used a multi-factor authentication schema for access control mechanisms and security controls. AlZain et al. [24] and Akshay et al. [25] conducted survey related to cloud security issues and addressed possible solutions for these issues.

## 3. THE PROPOSED ARCHITECTURE

Using public cloud servers to store client's data makes secure sharing of data a challenge. To protect confidentiality of data stored on the public server data access policies must be enforced. The cryptographic technique proposed in this work addresses this problem. Secret keys are kept by the data owner. Data must be encrypted before storing on a server, and the only way a client can access data s by supplying the corresponding decryption key. The main goal is to create a system that can defend itself from external and internal attacks. Architecture of the proposed cryptographic storage service shown on Figure 1 and consists of the following entities: data owner, client and cloud service provider.

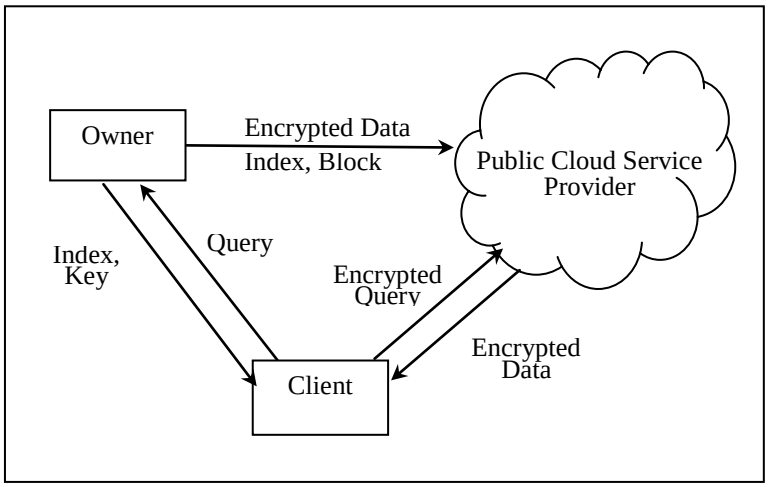


Figure 1. Architecture of a data outsourcing system

**Data owner:** wishes to outsource data onto a cloud provider server. The data owner module is responsible for encrypting the data. After encrypting the data by the proposed technique, the data owner sends data to cloud storage server. The data are stored in the public part of the data storage server. Data owner needs access to the cloud server only for loading and updating encrypted data.

**Clients:** trusted parties wishing to access the cloud data by means of encrypted identifier provided by the data owner. Only the legitimate clients are allowed to access and store data with confidence. To use the service the following tasks will be performed:

- Authentication of the client is conducted by the owner (user name, password) then session password sent to the client email or mobile phone.
- Then the client downloads an application which enables him to execute queries on the data without exposing the results of the queries to the cloud service provider.
- Using the application the client will be able to decrypt the cipher (encrypted original data) received from the cloud server. There is no need to download all the encrypted data, decrypt it and search locally. Clients wishing to search over the data generated by the owner (multiple readers/single writer) should run this application separately.

**Cloud Service Provider:** provides data servers and other computation and communication services on its premises. Encrypted data from data owners are stored on the data servers.
Proposed system enables a user with inadequate resources to search encrypted data at the public cloud server while protecting the privacy of the data. All the data have to be encrypted before

storage. Certain computations will be performed over the encrypted data to find out whether a block of encrypted data contains specific word(s) without disclosing any other information about the original text. The proposed system supports hidden queries that do not reveal the actual search word; controlled searching meaning no original words can be generated without the secret keys; and query isolation so the server cannot find out anything about the data and the search results.

A pre-computed index is built to speed up the searching. This index list all blocks containing key words in the encrypted database. It is generated by the client using one-way hashing. The system is capable of handling queries of different complexity

The system build of two main parts: owner and cloud service provider modules. The encryption of original data and computation and encryption of the index is performed on the owner side, and then this information is sent to public cloud service provider. The cloud service provider will build the index using the owner's encrypted index which points to the encrypted data at his site. The service provider cannot to derive any information from the encrypted data or from the index. Clients searching for blocks containing specific keywords encrypt the query at the owner's site and send it to the service provider's server. The search is conducted only on the index for the query, and the result is sent back to the client.

## 3.1 Owner's Module:

The owner's module employs encryption algorithm for each identifier associated with each block of data. Generated cipher is used to build an encrypted index and to encrypt each block of data associated with its index. Only a few secret keys needed to be maintained by the owner. Let's assume the owner chooses two secret keys: T and Y. The secret key T is used by the one-way hash function and Y to generate the index key for each block. Assume also that the outsourced data contain n identifiers with their corresponding blocks ($ID_1$, $B_1$; $ID_2$, $B_2$; ...; $ID_n$, $B_n$).

In this paper the author proposes to employ the one-way hash function to obtain a secret key $K_i$ for each identifier $ID_i$, use this key to generate a cipher for a corresponding block and corresponding encrypted index key. Thus, each block of data will be encrypted with a different secret key.

Two numerical values will be generated for each block $B_i$:

1) Encrypted index ($X_i$):

Compute $K_i = H_{ID}(T, ID_i)$, where $H_{ID}(T, ID_i)$ is collision-free one-way hash function. The hash function takes an arbitrarily long identifier as an input and output a short fixed length hash value. A hash value can serve various purposes. In our case it plays the role of an encrypted index or encrypted block of data.

This hash function works in the following way:

- Each identifier is partitioned into m blocks: $ID_{B0}$, $ID_{B1}$, … $ID_{Bm-1}$.
- The blocks then are encoded with the multi- hashing block mode shown in Figure 2.
- The final hash value is:

$$H_{ID} = T_{IDm-2} \oplus H_{IDBm-1} = (T_{IDBm-3} \oplus H_{IDBm-2}) \oplus H_{IDBm-1} = \ldots$$
$$= T_{IDBm-1} \oplus \ldots \oplus T_{IDB1} \oplus (T \oplus T_{IDB0}), \quad \text{(Eq.1)}$$

where $\oplus$ is the bit-wise XOR operation.

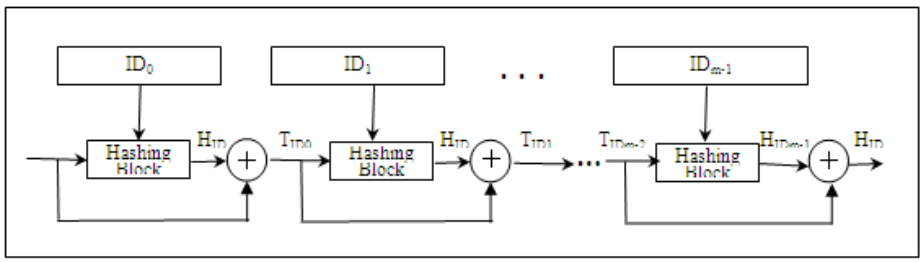


Figure 2. The Multi- Hashing block

2) Encrypted block ($S_i$):

Compute $S_i = K_i \oplus B_i$, where $S_i$ is the encryption of $B_i$ ($S_i = H_{ID}(T, ID_i) \oplus B_i$).
Note that there is no need to store any encryption keys neither $X_i$ nor $S_i$ on the owner site.
The data owner responsibilities include encryption of data and different data operations. He may need to perform various data structure operations on data records such as insert, update, delete. The mechanisms to handle these operations are below.

**Insert:**

The insert operation is similar to the procedures used to prepare encrypted data for loading. To insert a block $B_i$ with its associated identifier $ID_i$, compute the following:

$$nX_i = H_{ID}(T, ID_i) \oplus Y \qquad \text{(Eq.2)}$$
$$nS_i = H_{ID}(T, ID_i) \oplus B_i \qquad \text{(Eq.3)}$$

The owner then sends insert ($nX_i$, $nS_i$) to the service provider which will update the index tree to reflect the insertion.

**Update*:***

To modify the outsourced block of data from $B_i$ to $D_i$ the owner will use the original key to encrypt $D_i$ by computing: $dS_i = K_i \oplus D_i$ and send update ($X_i$, $dS_i$) to the cloud service provider who will update the encrypted block pointed by $X_i$.

If a client wants to update data, he has to obtain the write privilege from the owner. To update a block a client he must send updated data to the owner to get the new encryption and index keys.

**Delete:**

To delete a data block $B_i$ from the outsourced data, the owner should calculate its encrypted index $X_i$ and its encrypted block $S_i$; then send the operation delete ($X_i$, $S_i$) to the cloud service provider. The cloud service provider has to locate to be deleted node in the index tree ($X_i$), and then remove it from the index and remove its associated encrypted block ($S_i$).

### 3.2 At the cloud service provider site:

As volume of organizational becomes extremely large, it is preferable to keep the size of outsourced data as small as possible, meaning organizations should store on the cloud server only one copy of each encrypted data block with its associated index.

The cloud service provider receives from the owner two objects for each block of data $B_i$: encrypted block $S_i$ and encrypted index $X_i$ of the identifier $ID_i$ both generated using symmetric key K, and block $B_i$ identifier $ID_i$

Encrypted values of keys Xs are used to build indexes. Each encrypted index serves as a pointer to its associated encrypted data block (Figure 3).

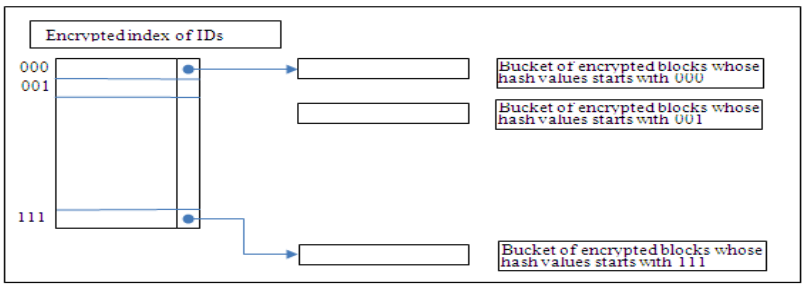


Figure 3. Encrypted index

The index can be looked at as a binary digital tree where search is executed on a bit-by-bit basis. To begin with, the first group of index key values(X) with their encrypted blocks (S) is inserted into one page. If the page overflows, it is split into 2 pages on the basis of the high-order bit of the key. All keys having a value of zero on the high order bit are placed (with their blocks) in one page and those having the value of one, with their blocks, are placed on the second page. As these changes occur, the directory has to be updated. This process can continue until all bits of the key are used. The directory building approach is a digital binary tree where each node has 2 pointers. An internal node has pointers to the 2 children, while an external node has a zero on the left (dummy page) pointer and a pointer to the leaf page referenced. A binary tree with six leaves and five internal nodes is shown on Figure 4. The benefit of using this kind of tree is that searching becomes much simpler.

The following actions must be performed to determine which page the identifier belongs to:

- Starting from the root, key is checked bit by bit from the left.
- Each time a 0 is encountered go left, a 1 - go right.
- The process continues until a leaf is reached, and returns the encrypted result to the client side.

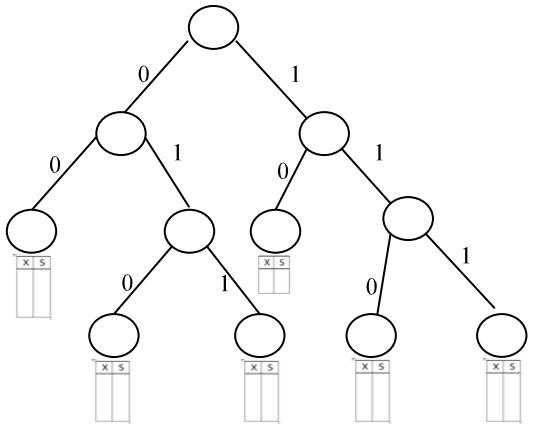


Figure 4. Binary digital tree

### 3.3 At the Client Site:

Clients are allowed to access only data they need to protect confidentiality. A client sends a request (an identifier) to the owner to get the encrypted key. The owner will compute the encryption key $K_i$ and its index $X_i$ utilizing a one-way hash function and send both back to the client over the secure communication channel. The client will communicate directly with the cloud service provider and present the index $X_i$, which will forward the corresponding encrypted data $S_i$ referenced by $X_i$ to the client. Locally the client calculates: $B_i = S_i \oplus K_i$ to generate the original block of data.

## 4. SECURITY ANALYSIS

The proposed technique insures that the cloud administrator does not unintentionally violate cloud users' data security. It guarantees confidentiality of the outsourced data against unauthorized users and a curious cloud service provider. The proposed architecture will allow only the legitimate client to access and store data with confidence. Data are encrypted before storage in the cloud using a symmetric encryption algorithm. An encryption key $X_i$ is generated for each data block $B_i$, thus there are n keys for n blocks of data. Each data block's private key is owned by the owner of the data, so that no one else can decrypt the data. This solution can protect the data not only from intruders, but even from the cloud service provider's staff curiosity, who cannot learn anything of the owner's outsourced data contents, or the client's queries performed on the server, or the returned results, thus protecting clients' privacy. Queries are executed on the encrypted data sets. Clients can be sure that data returned from the public cloud server originated from the data owner and has not been tampered with. Clients are not allowed to access data outside of what they are querying. To access cloud data, a user has to be granted privilege by the owner with respect to the private keys related to the subject of his search. After authentication client can down load data from the cloud and decrypt the data locally.

It is infeasible to find an identifier or original data that has a given its key index value ($Xi = K_i \oplus Y$), or the encrypted block of data ($S_i = ht (T, ID_i) \oplus B_i$) because of the security property of one-way chaotic hash function. It is very difficult to derive the key Y from the value of $X_i = K_i \oplus Y$. It is infeasible to compute the original values of $X_i$ or $S_i$, because it is computed by the secure one-way chaotic hash function. Changing even one bit of the hash input will cause huge changes in the final hash value due the sensitivity of the chaotic hash function. The proposed technique

requires less computation cost. Most of computations implemented by using exclusive OR and multiplication thus require much lesser computation power than costly modular exponentiation techniques. The proposed technique is communication secure and efficient: It provides protection in data during the transmission, and due to the use of fixed key-size ($X_i$) from the user to the cloud service provider and from cloud service provider to the user ($S_i$), each message-size is reduced to fixed number of bits.

## 5. CONCLUSION

The problem of security of outsourced data in public cloud storage has been examined in this study. Developed efficient cryptography mechanism allows to achieve secure and efficient access to cloud data. The owner is in control of the data, and no actions of the cloud service provider can compromise the confidentiality of the data. The developed encryption schema offers a level assurance that only authorized users can see the data. As the result, users can conduct secure search on encrypted data in a public cloud infrastructure. An efficient secure index for each identifier has been developed utilising one-way hashing functions. In this study each data record is encrypted with a different symmetric key generated from the secure index so that flexible cryptography-based control can be accomplished; and confidentiality of the outsourced data against the cloud service provider and unauthorized users is guaranteed. Other advantages of the proposed system are minimizing communication overhead and computations on both client and server sides and thus outsourcing has the potential to minimize client-side management expenses and benefit from a service provider's expertise.

The main focus of this paper is security of data in case of a public cloud service provider. Two more issues need to be looked into. One of them is the issue of an untrusted client, and another is outsourcing data and data owner's application software to a cloud service provider, where security of the application code have to be assured as well as the security of the outsourced data.

## REFERENCES

[1] Dakhane, D.M. & A.A. Arokar, (2012 ) "Data Secuity in Cloud Computing for Biometric Application," International Journal of Sci Research, Vol. 3, No. 6, pp. 1-4.

[2] Cloud computing use case discussion group, "Cloud Computing Use Cases: White Paper" available from: http://cloudusecases.org, 2012

[3] Al-Sakran, Hasan Omar, Bin Muhaya, Fahad & Irina Serguievskaia, (2011) "Efficient Cryptographic Technique for Securing and Accessing Outsourced Data", International Journal of Computer Science and Information Security, Vol. 9, No. 8, August 2011.

[4] Yang, Huaqian ;Wong, Kwok-Wo; Liao, Xiaofeng; Wang, Yong & Degang Yang ,(2009) "One-Way Hash Function Construction Based on Chaotic Map Network", Chaos, solutions & fractals, Vol. 41, No. 5, pp. 2566-2574.

[5] Wang, W., Li, Z., Owens, R. & B. Bhargava, (2009) "Secure and Efficient Access to Outsourced Data", in Proc. of ACM Cloud Computing Security Workshop, pp. 55-65.

[6] di Vimercati, S. D. C., Foresti, S., Jajodia, S., Paraboschi, S. & P. Samarati, (2007) "A data outsourcing architecture combining cryptography and access control", in Proc. of the ACM workshop on Computer Security Architecture, pp. 63-69.

[7] S di Vimercati, S. D. C., Foresti, S., Jajodia, S., Paraboschi, S. & P. Samarati, (2007) "Over-Encryption: Management of Access Control Evolution on Outsourced Data", in Proc. of the International Conference on Very Large Databases, pp. 123-134.

[8] Yin, X. Z.; Liu, H. & Jae Lee, (2014) "An Efficient and Secured Data Storage Scheme in Cloud Computing Using ECC-based PKI", IEEE 16th International Conference on Advanced Communication Technology (ICACT) , pp. 523 – 527.

[9] Yang, Ching-Nung & Jia-Bin Lai, (2013) "Protecting Data Privacy and Security for Cloud Computing Based on Secret Sharing", International Symposium on Biometrics and Security Technologies (ISBAST), pp 259 – 266.
[10] Ullah, S. & Z. Xuefeng, (2014) "TCLOUD: A Trusted Storage Architecture for Cloud Computing", International Journal of Advanced Science and Technology Vol.63, pp.65-72
[11] Dang Tran Khanh, (2009) "Security issues in outsourced xml databases". In Open and Novel Issues in XML Database Applications: Future Directions and Advanced Technologies. IGI Global,.
[12] Song, D.; Wagner, D. & A. Perrig, (2000) "Practical Techniques for Searches on Encrypted Data", in Proc. of the 2000 IEEE Symposium on Security and Privacy (S&P 2000).
[13] Chang Y.-C. & M. Mitzenmacher, (2005) "Privacy preserving keyword searches on remote encrypted data", ACNS (2005).
[14] E. Goh, (2003) "Secure indexes", Cryptology ePrint Archive, Report 2003/216 (2003), http://eprint.iacr.org/2003/216
[15] Golle, P; Staddon, J. & B. Waters, (2004)"Secure conjunctive keyword search over encrypted data", in M. Jakobsson , M. Yung and J. Zhou, editors, (ACNS 2004), LNCS, Springer, Heidelberg: 2004, 3089, pp 31–45.
[16] Brinkman, R.; Feng, L.; Doumen, J.M., Hartel, P.H. & W. Jonker, (2004) "Efficient Tree Search in Encrypted Data", 2nd International Workshop on R. Security in Information Systems, April 2004.
[17] E. Goh, (2003) "Building Secure Indexes for Searching Efficiently on Encrypted Compressed Data", http://eprint.iacr.org/2003/216/
[18] Boneh, D.; Crescenzo, G. D.; Ostrovsky, R. & G. Persiano, (2004) "Public-key encryption with keyword search", In: C. Cachin, editor, Proceedings of Eurocrypt 2004, LNCS, Springer-Verlag, May 2004.
[19] Dai, J. & Q. Zhou, (2010) "A PKI - based Mechanism for Secure and Efficient Access to Outsourced Data", 2010 International Conference on Networking and Digital Society.
[20] Yin, XiaoChun; Lui, ZengGuang & Hoon Jae Lee, (2014) "An Efficient and Secured Data Storage Scheme in Cloud Computing Using ECC-based PKI", IEEE 16th International Conference on Advanced Communication Technology (ICACT), pp 523 – 527.
[21] Yang , Kan; Jia, Xiaohua; Ren, Kui & Bo Zhang, (2013) "DAC-MACS: Effective data access control for multi-authority cloud storage systems", INFOCOM, 2013 Proceedings IEEE , pp 2895 - 2903
[22] Ruj, S.; Stojmenovic, M. & A.Nayak, (2014) "Decentralized Access Control with Anonymous Authentication of Data Stored in Clouds), IEEE Transactions on Parallel and Distributed Systems, pp 384 – 394.
[23] Bamiah, Mervat Adib; Brohi, Sarfraz Nawaz; Chuprat, Suriayati & 2Jamalul-lail Ab Manan, (2014) "Trusted Cloud Computing Framework For Healthcare Sector". Journal of Computer Science Vol. 10, No 2, pp 240-250.
[24] AlZain, M.A.; Soh, B. & E. Pardede, (2013) "A Survey on Data Security Issues in Cloud Computing: From Single to Multi-Clouds", Journal of Software, Vol. 8, No. 5, May 2013
[25] Kapse, Akshay D. & Piyush K. Ingole, (2014) "Secure and Efficient Search Technique in Cloud Computing", Fourth International Conference on Communication Systems and Network Technologies, pp 419 – 429.

**Author**

**Dr. Hasan Al-Sakran** currently is an Associate Professor in the Department of Management Information Systems at the King Saud University/Saudi Arabia. He has a D. Sc. degree in Information Systems Design from the George Washington University/ Washington DC. His research interests include: Agent Technology applications, E-commerce applications, Security of Information systems, Case-Based Reasoning, Software cost Estimation.