I.J. Modern Education and Computer Science, **2015, 5, 16-23**
Published Online May 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2015.05.03

A Framework for Homomorphic, Private Information Retrieval Protocols in the Cloud

Mahmoud Fahsi, Sidi Mohamed Benslimane
EEIDIS Laboratory, Djillali Liabes University, Sidi Bel Abbes, 22000, Algeria.
Email: {mfahci, benslimane}@univ-sba.dz

Amine Rahmani
GeCoDe Laboratory, Taher Moualy University, Saida, 20000, Algeria.
Email: aminerahmani@univ-sba.dz

***Abstract*—**Professional use of cloud health storage around the world implies Information-Retrieval extensions. These developments should help users find what they need among thousands or billions of enterprise documents and reports. However, extensions must offer protection against existing threats, for instance, hackers, server administrators and service providers who use people's personal data for their own purposes. Indeed, cloud servers maintain traces of user activities and queries, which compromise user security against network hackers. Even cloud servers can use those traces to adapt or personalize their platforms without users' agreements. For this purpose, we suggest implementing Private Information Retrieval (PIR) protocols to ease the retrieval task and secure it from both servers and hackers. We study the effectiveness of this solution through an evaluation of information retrieval time, recall and precision. The experimental results show that our framework ensures a reasonable and acceptable level of confidentiality for retrieval of data through cloud services.

***Index Terms*—**Cloud health Storage and Retrieval, Private Information Retrieval, Cloud User Privacy, access control

I. Introduction

Textual data was once housed under user supervision, and now has been placed under service provider protection, for example in data cloud servers and Data-Warehouses. Users have lost control of their data, as their data are no longer stored under their watchful eyes. Although third-party servers cannot access users' data, owing to usage license agreements, query receiving and processing remain insecure, as third-party server administrators have all permissions to access both queries and user actions log files [1].

Since the web was first created, users have viewed it as a cloud of servers that contains data and applications. However, cloud computing and storage perception still a new concept [2]. To help clarify our paper, we will provide an overview of the cloud computing evolution. Also, we explain the Privacy preserving interest for cloud computing regarding access control ethic and encryption methods as PIR protocols.

Moreover, the cloud-computing is recognized as a new outsourcing processing and data storage technologies purchased as a customized service with no need to pay for hardware and software platform on the client side. Cloud almost important idea is "pay for what you use"; the main goal was to create a reliable, fast, and economical technology [3]. The discussion concerned renting storage space as a service within the Software as a Service offer (SAAS), using remote pre-installed software applications within that space without the need to install them on distant servers. Otherwise, Cloud technology may give their user the possibility developing and deploying their own applications using platforms as a service (PAAS).

However, security issues are the biggest impediment to the growth of this technology. Moreover, some data have much more stringent security requirements than other data. To address security issues, ways of improving all user services, from data storing to retrieving and downloading information, have been proposed. To successfully implement a private retrieval operation that preserves confidentiality, authenticity and integrity, we can use the Private Information Retrieval (PIR) protocols [4], to unable any authorized server administrator and to access retrieved information expressed in query language form or the resulting ranking given by the retrieval algorithm. The PIR protocols are implemented in real software systems, using various cryptographic algorithms. Although many PIR protocol implementations exist in the literature [5], we will use the homomorphic PIR protocol, based on asymmetric encryption [6].

The main objective of this paper is to introduce an Information Retrieval module into the existing cloud storage architecture, using PIR to address privacy concerns regarding critical textual data stored away from the user's supervision. Using basic Information Retrieval indexing methods, such as term frequency and inverted term frequency TF/IDF implementation in terrier, as an input for our encryption system, we will vary six (06) different homomorphic PIR algorithm to study encryption time and retrieval effectiveness.

The paper proceeds as follows: in section two, we provide a highlighted survey of secure retrieval models already implemented through cloud storage computing. In

section three, we give a general view of the present a number of definitions and basic concepts used in this paper. Section four describes a practical application that explains our work. Next, the architecture and implementation sections describe our contribution in detail. A final section presents our conclusions and discusses future research.

II. Related Work

Many related papers treat the data security issue but focus on encryption algorithms, transport protocols and user strategies for accessing related data stored in the cloud. All of these measures guarantee to all enterprises using the cloud the privacy of the user's data based on the idea of storing data in encrypted form. To describe the security involved, the security federation [7] explicitly lists eight risks that any Cloud user should know about before using cloud services. The most common risk is losing control over one's data while they are in the cloud. The second most common risk is service non-availability caused by server breakdown. Kevin Hamlen and al. [8] discuss the storage security and data layers in Cloud Computing and, additionally, describe another problem related to large data volumes, namely, that searching for information can be similar to searching for a needle in a haystack.

Carlos Aguilar Melchor [9] describes the PIR protocols as a way of retrieving information while concealing retrieved information from the server. Adi Shamir [10] also describes the basic approach of exchanging data, using a third-party trusted actor responsible for generating encryption and decryption keys for homomorphic encryption. This work was the original PIR protocol description. The main idea behind this proposition is that the owner of documents must index them and then encrypt them and send them to another client. The owner can give permission to other users to retrieve the information from his account by providing them with the secret and public keys. This work was later modified by Chor and al. [4] in such a way that both documents and indexes are encrypted and stored on the server-side for the next retrieval step.

Only a few papers on securing user data access via the cloud storage access procedure have been available because this domain is new, and nearly all security initiatives concern the need for more restrictive institutional legislation to control cloud administration due to the lack of privacy [11]. Similarly, new bugs and vulnerabilities targeting the cloud are proliferating [12]. Large companies' networks offer more targets for hackers. Cloud suppliers, which are often larger than their clients, are also attractive targets, as the cloud offers a high "surface area of attack" [13]. A commonplace observation is that, while cloud providers offer advanced services, their performance with respect to privacy and security has been weak [1], [14].

X. Zhang and al. [15] propose a new method of indexing data in cloud based on quasi-identifier indexing. The originator presents data in incremental form and then sends the data to the healthcare cloud provider. As in Shamir's contribution, the indexed data will be stored in encrypted form. The purpose of the paper is to resolve the problem of retrieving huge amounts of decentralized data stored in the cloud while ensuring greater scalability and efficiency in searching encrypted data.

Melvin Greer and John W. Ngo [16] present and describe flexible search functionalities of encrypted data as a new approach to searching for data within the cloud, an idea shared with Cong W. and al [17] in the use of ranked keyword search to retrieve data. However, this paper proposes a way of storing data in encrypted form, such that a user can give permission to others to search through his data by sharing keys. This could be done using other channels of the cloud service itself.

Khan and al. [18] present a method for secure mobile access to Cloud computing facilities. The authors guarantee users of mobile devices, especially smart phones, a secure cloud storage service accessible via a mobile device, proposing an architectural framework based on these six operations: registration, redirect registration, user credentials, redirect credentials, upload and download using certification and authentication mechanism. In this case, data encryption during indexing and retrieval is performed by a third-party.

In PIR practicality experiments, Sion and al. [19] and Olumofin and al. [20] show that single database PIR computation is more than one order of magnitude slower than trivial data transfer computations. Additionally, multi-server PIR computation is more practical, although it requires that servers do not collude.

Based on Sion's and Olumofin's results, we propose to join every data encryption with user access policies instead of with new queries, so that the public and private keys are changed only when the password and user-names change. On this basis, encryption will be performed once per month or once per week, depending on confidentiality policies.

III. Motivation and Problem Analysis

Consider a scientific study of a hospital cloud storage service used for information collection, obtaining online answers to questions about specific symptoms and medical diagnoses — evolving in time and that requires users to provide a large amount of personal information.

The impediment of storage cloud services is that users lose control of their data once they enter the cloud, which makes privacy an obstacle and a goal at the same time. The goal of privacy is relevant not only to user information security but also commercial success.

The first solution to the privacy issue that comes to mind is to store data in encrypted form, which requires that the encryption and decryption keys be fixed from the beginning. This may at first appear to present an optimal solution.

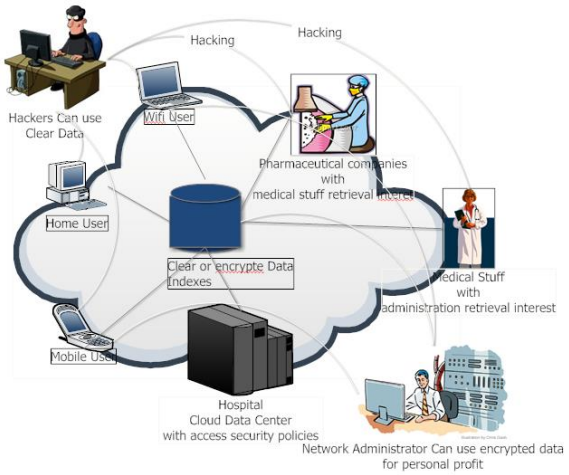

Fig. 1. Health storage cloud service

However, as noted above, the problem of hackers leads to another problem, namely, the question of whether to conduct operations on encrypted data, ensuring privacy, or decrypt the data, making it impossible to ensure privacy at the server.

Viewed differently, if a client is hacked and his data are stolen, hackers can use the keys to retrieve data in the cloud. Then the server will view the hacker queries as legal, which is the response that the hacker wants.

The system we propose is based on the idea of altering encryption keys with every query, using a Private Information Retrieval Protocol with specific attributes. Encryption then depends on the username and password attributes. We also propose periodic personal data encryption with each new user query, where such encryption uses a new predicate generated from the user information.

IV. Preliminaries

Homomorphic encryption is a cryptographic concept developed to address the need to be able to execute operations on ciphered data without decrypting them. This encryption system is homomorphic only if it has the follow property: where "Enc" is an encryption equation, and x and y are plain text. Homomorphic encryption is semantically secure if it reveals no information about x and y and it is impossible to distinguish between x≠ y and x=y in the encryption equation [6].

A. Predicate Based Encryption

The predicate-based encryption (PBE) protocols are a specific type of asymmetric encryption system that uses one or a set of predicates corresponding to the cipher side algorithm to generate encryption and decryption keys.

Viewed differently, the PBE algorithm consists of introducing authentication and access controls into an encryption system [5]. The PBE scheme is based on four principal parts and works as follows;

Setup: This procedure initializes the encryption process and generates 2 keys: the Master Secret Key (MSK) and a set of Master Public Parameters (MPK).

$$Setup\ () \rightarrow (MPK, MSK) \quad (1)$$

Keygen: This procedure consists of reading the master secret key generated in the setup and the predicates, with the objective of generating the secretive key (SK) corresponding to the entity.

$$Keygen\ (MSK, Predicates) \rightarrow Sk \quad (2)$$

Encrypt: This is the encryption procedure. It uses public parameters generated in the setup to encrypt a plaintext m to obtain a ciphered text (CT).

$$Encrypt\ (MPK, m) \rightarrow CT \quad (3)$$

Decrypt: The decryption procedure decrypts the ciphered text by using the secret key generated in the Keygen step.

$$Decrypt\ (Sk, CT) \rightarrow m \quad (4)$$

There are a several variations of the PBE; some of them use a supplied encryption key (use-key) corresponding to an entity to encrypt messages.

$$Encrypt\ (MPK, m, use\text{-}key) \rightarrow CT \quad (5)$$

Others add a new procedure, called "delegate," which consists of delegating the decryption side of an entity to another entity. PBE schemes are of several types, depending on the use predicate. Melchor and al. present three types of PBE in their paper [5], as described next.

Quadratic Residuosity:

For attribute selection, we must consider p and q as two integers.

$$X2 \equiv p \bmod q \quad (6)$$

We say that p is a quadratic residue of q if and only if there is an integer x that verifies the property cited above.

B. Identity Based Encryption (IBE)

This type of PBE uses just one type of information, such as the user's identity, for example, one's email address or user name or a concatenation of information to form a single attribute. Adi Shamir cited a first version of an identity-based cryptosystem in his paper "Ciphertext-Policy Attribute-Based Encryption" [10], and other as do Boneh and Franklin [21] improve the initial work by imposing a variety of rules on identity information selection and combination.

C. Attributes Based Encryption (ABE).

This type of PBE uses a set of attributes to generate secret keys, where the attributes do not necessarily correspond to the user entity. Such schemes can thus use, for example, the protocol employed in the transmission (http, FTP…), port numbers such as TCP/IP or other information described by Goyal and al. [22] and Bettencourt and al. in [23].

D. Specific Attribute-Based Encryption.

This type of PBE is similar to IBE. However, in this case, the cryptosystem employs a specific attribute during construction by supporting certain mathematical characteristics such as inner products, used by Katz, Sahai, Waters and al. [24], or hidden vectors, used by Boone and Waters [25].Thus, these cryptographic systems can be referred to as the predicate used in the key generation step.

V. System Architecture

A. Main Idea

In our system, a health cloud storage service provider allows clients to store and retrieve medical documents and information. Thus, users, similarly to the patients, place their medical reports and data in an expanding database that medical staff can access and thereby follow medical developments from home.

Additionally, the research center can take information about experiments and place the results, including confidential details, into storage. Once a document is sent to the cloud, it will be indexed using terminological indexation. Each document will be stored in this index, and all indexes will update the global index.

The goal of using indexation is to accelerate the retrieval process. Now, if someone wishes to diagnose his medical condition, he must find the necessary information about his illness by generating a query containing confidential details such as his name, age, country and so forth.

Our proposal is to encrypt client queries and data in the cloud with the same public key, using a variation of Private Information Retrieval Protocols within the same framework. The keys in this case are temporary, and the server does not know the secret key. Thus, the retrieval process is performed on encrypted data, and returned results remain encrypted until they are received by the client who possesses the secret key to decrypt them.

The stored document on the cloud explicates the hacker major interest. For example, any cloud storage that contains information available in an encyclopedia, such documents does not represent a target for hackers. The goal of attacking such a service is to obtain confidential information contained in queries and retrieved results.

To this end, the adversaries violate the confidentiality of the retrieval process through phishing attacks or by placing sniffers at service providers. Our proposed solution is thus effective in protecting users' private information revealed in queries and in retrieved documents.

B. Framework description

We sought to implement an initial Private Information Retrieval framework in a cloud storage system based on two principal modes: the connection and inscription mode, used to guarantee authenticity, and the storage and retrieval mode, used to ensure confidentiality. In the next subsection, we will examine each mode separately in greater detail.

C. Connection Inscription mode;

This mode is a simple human-machine interface for new user subscriptions and existing user connections. The main objective in this case is to enable the right authorizations to be given to users, with the goal of protecting access through an authentication mechanism.

Figure (2) illustrates the subscription method in our system; to subscribe, a doctor or a patient must give some information (first name, last name, username and password) to the principal server (service provider), which in turn must verify the existence of the username. In other words, the server determines whether the username is already used by another user. Thus, in the figure, "yes" means that the username is already in use, and "no" means that it is not.

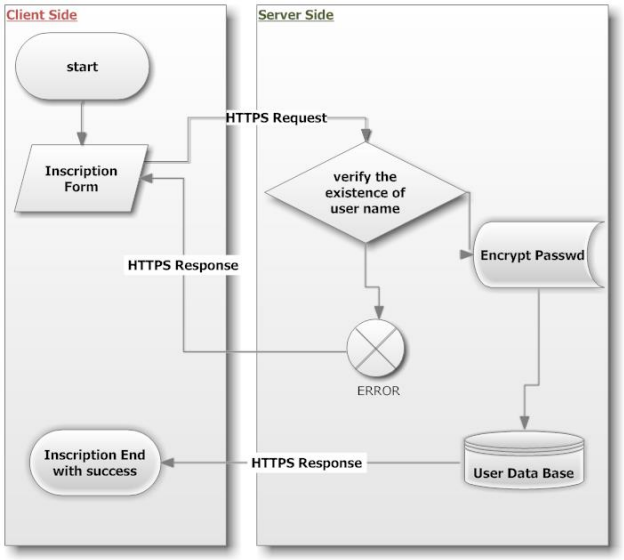

Fig. 2. inscription method

If a username is already in use, an error message will be returned to the user. Otherwise, the password will be hashed, and all information will be stored in the user's Database. The purpose of hashing the password is to protect users' data against SQL Injection attacks. Even if the user's Database is stolen, the retrieved information will not contain unknown information because the hashed function proprieties are unidirectional. Now, if the user is already a subscriber, he must connect to use the document storage retrieval service. For that, the user must send his username and his password, as illustrated in figure 3.

When the cloud server receives the user's request, it will verify the existence of the username in its database before allowing the user to access the storage retrieval interface. If the username cannot be verified, the system returns an error message to the user.

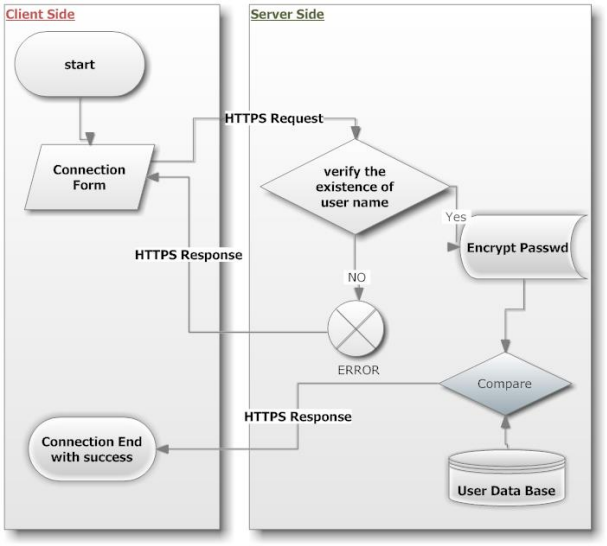

Fig. 3. Connection mechanism.

D. Storage Retrieve mode;

Entrance via this mode implies that the patient is already a subscriber, is authenticated and connected to the service through his account and can conduct uploading and retrieval operations on documents, according to his rights within the system. The user can also be a medical researcher who wishes to obtain information about the patient population.

As figure 4 shows, the storage mechanism has no security issues except the protection of transmissions between the user and server achieved through secure web protocols (HTTPS, SSH…). Thus, when the user uploads textual documents to the cloud data center, the server indexes those documents to facilitate further retrieval operations.

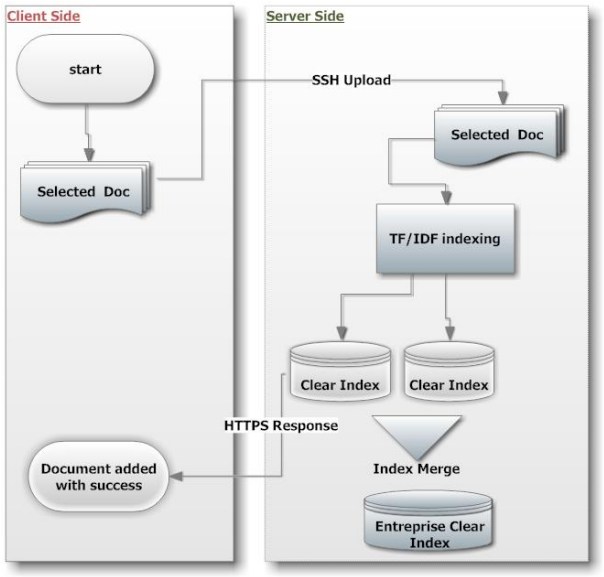

Fig. 4. Storage mechanism

If the storage process works well, a "success" message will be returned to the user. In cases of indexation or storage errors, the system returns an error message to the user.

Finally, the cloud server uses the public key to encrypt the user's data index (or cloud index, in the case of a doctor's query) at his datacenter and applies TF/IDF matching to the encrypted query and encrypted indexes to obtain an encrypted result that is returned to the user via an https response. The user machine uses the corresponding secret key to decrypt the results for further selection or downloading.

To test the efficiency of our proposed solution, we sought to implement a set of PIR protocols on the same data corpus, using the same indexing method (Retrieval Model). This implies that different cryptographic algorithms are used to evaluate encryption time and the pertinence of each protocol.

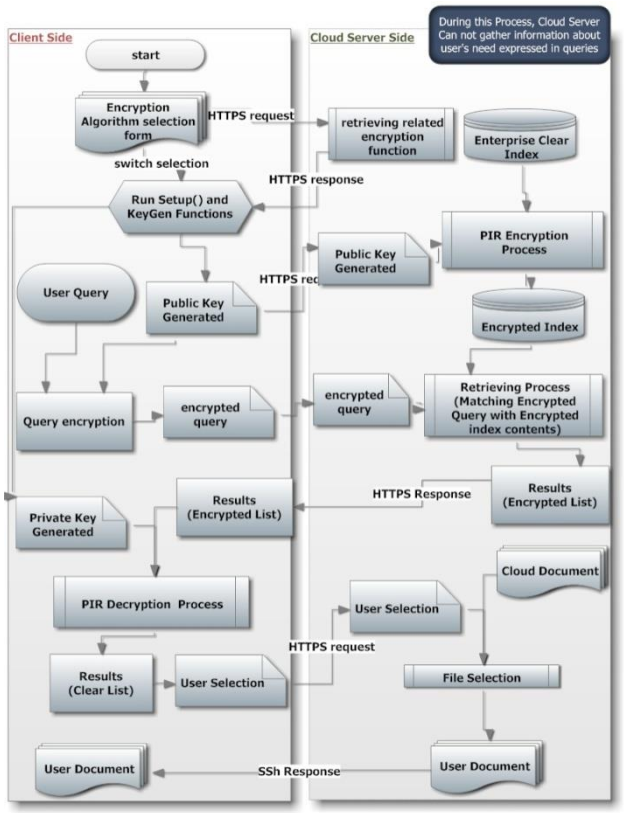

Fig. 5. Retrieval process.

VI. Experimentation

All algorithms used are homomorphic, although some are PBE algorithms, while others are not. The following table illustrates the proprieties of each algorithm that we used. We conducted our tests on a local server with a 2.5 quad core processor, 8 GB of RAM and 256GB SSD hard drive.

We choose to implement all algorithm using terrier information retrieval platform configured with basic indexing method witch create a compressed index. Our algorithm will index the document will encrypt term frequency keyword only after frequency creation which allow as to enormously reduce the encryption time.

Table 1. Used Algorithms Types.

	Homomorphic algorithm	PBE algorithm
TSZ	YES	YES
Waters	YES	YES
Okamoto-Uchiyama	YES	-
Paillier	YES	-
Goldwassers-Micali	YES	-

The TSZ and Waters PBE algorithms were implemented because they both utilize user identity to generate keys, so that the keys will change according to passwords. Ultimately, identification is ensured through the use of these two algorithms.

Additionally, every protocol has its own encryption algorithm. This is notable in the case of Waters because of the way it makes data decryption impossible if the encrypted form is not a quadratic residue of the secret keys. The Waters algorithm thus ensures not only confidentiality and authenticity but also the integrity of the results.

The following table indicates, for each algorithm, the security issues that it addresses.

Table 2. Security mechanism addressed by PIR cryptographic algorithms.

Algorithms	Privacy	Integrity	Key generation based user's identifier
TSZ	YES	-	YES
Waters	YES	YES	YES
Okamoto-Uchiyama	YES	-	-
Paillier	YES	-	-
Goldwassers-Micali	YES	-	-

Our framework was implemented and tested on the MEDLINE corpus developed by the U.S. National Library of Medicine NLM. This corpus is composed of 1,033 citations of medical articles in textual form.

Encryption key sizes were fixed at approximately 1O24 bits, and the query size was approximately 304 to 408 bits in plain text form and 1040 bits in encrypted form. The next step was to compare the protocols in terms of encryption time, retrieval time and returned results pertinence (recall, precision and F-measure).

Table 3. Queries size

No.	Query	Length in bits
01	The crystalline lens in vertebrates includes humans	408
02	Electron microscopy of lung or bronchi	304
03	Tissue culture of lung or bronchial neoplasm	360

We performed our tests by 3 short queries selected from the MEDLINE collection. All next results are the average of the respective results related to those queries.

VII. Results and Discussion

Because our interest lies in the private retrieval protocol adaptation of the cloud architecture, our experimental results, presented in table (4) were obtained for a fixed query size.

Table 4. Recall, precision and F-measure

Algorithms	f-measure	Recall	Precision
Clear	0,73	0,68	0,81
TSZ	0,78	0,88	0,70
Waters	0,78	0,68	0,92
Okamoto-Uchiyama	0,87	0,93	0,82
Paillier	0,78	0,88	0,70
Goldwassers-Micali	0,72	0,88	0,61

The best result among all protocols was shown by the protocol using the Okamoto-Uchiyama algorithm, with an F-measure value of 0.87. Otherwise, The Waters algorithm exhibited the highest precision.

Goldwassers-Micali displayed the poorest performance, with an F-measure of 0.61. Figure 6 graphically compares recall and precision for each algorithm. As noted above, our protocols will encrypt only Inverted index and related query to retrieve information.

We compared protocols in terms of encryption and retrieval. Table (5) lists retrieval process execution times, providing some idea of which is the fastest protocol.

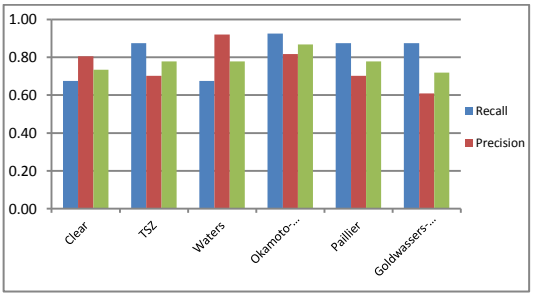

Fig. 6. Graph recall and precision by protocol.

Execution time is a very important evaluative criterion of a cloud service (users must not wait long to receive desired responses).

The reason for the large values of the encryption and decryption times of most protocols is the large size of both the encryption key and the query used for the encryption and decryption processes (the encryption process key was represented in java by Big Integer of 1024 bits and the encrypted query size was represented by 1040 bits).

Table 5. Retrieving Time

Algorithms	Encryption Time(sec)	Retrieval Time(sec)	Total (sec)
Clear	0,00	0,02	0,02
Waters	0,10	0,10	0,20
Paillier	0,37	0,04	0,41
Okamoto-Uchiyama	1,04	0,01	1,05
TSZ	1,08	0,03	1,11
Goldwassers-Micali	1,36	0,01	1,37

As seen in table 5, the Waters algorithm is the fastest, while the Goldwassers-Micali algorithm is the slowest. On the other hand, execution time is very important, due to the cloud machine specification used to implement our framework.

As with any framework, our system has some limitations, each suggesting possible future solutions. The directions for our future research are likely to improve our proposed system. Initially, with each receipt of a new query, encryption involves changing the encryption and decryption keys with every new query.

This resolves the issue of user privacy but creates a significant computation problem, due to the quantity of documents the user may store in the cloud. This is why, in the future, we will implement a different key-generation strategy and study its effect on access and execution times.

Additionally, we will explore other Information Retrieval models for greater index compression that will improve the effectiveness of our framework. In addition, we can implement other types of Private Information Retrieval (PIR) algorithms and study their compatibility within the cloud framework.

VIII. Conclusion

As a technology, cloud storage and computing has its own properties, some of which are highly attractive to companies and users. In general, users are interested in this service if all their access and retrieval actions through the cloud providers can be guaranteed by companies. For this reason, we studied a future extension involving both a retrieval model and a secure or Private Information Retrieval Protocol as an aid to the effective use of enterprise data stored in the cloud.

According to our hypothesis, stored information is not the only target for hacker and competitive companies. Exchanged data are very important for building a real view on user needs or common company's service. Therefore, we have created a set of Private Information Retrieval (PIR) protocols to retrieve data without revealing any information about cloud user queries or results that are usually captured by the cloud administrator on the server interface or by hackers on the transmission channel.

After many experiments and evaluations of results, we can confirm that our system can be used to manage data in sensitive domains such as health care and other fields. In this evaluation, we have highlighted the fact that our system ensures an acceptable execution time with high level of confidentiality in retrieving data process through cloud storage services.

We have also verified the results of Sion and Olumofin and avoided the time obstacle by joining encryption parameters with access policies rather than users' queries. In this case, we compared multiple decryption process results in term of time and we concluded that all used algorithms gave almost the same decryption time. With respect to the recall and precision results affected by encryption key and queries size, additional improvements can be made by adding a query size limit to minimize the role of randomness in the results.

References

[1] Wittow, M. H., Buller, D. J., Cloud Computing: Emerging Legal Issues for Access to Data, Anywhere, Anytime. Journal of Internet Law, Jul, 14(1), 1-10. (2010).

[2] Harfoushi, O., Alfawwaz, B., Ghatasheh, N. A., Obiedat, R., Abu-Faraj, M. M., Faris, H., Data Security Issues and Challenges in Cloud Computing: A Conceptual Analysis and Review. Journal of Computer Science & Communications, PP. 15-21, (2014).

[3] Mark, P., Jason, B., Monya, K., White Paper: Cloud Storage: the Next Frontier for Tape. Enterprise Strategy Group, commissioned by Oracle and is distributed under license from ESG. (2013).

[4] Chor, B., Goldreich, O., Kushile, E., vitz, and Sudan, M., Private Information Retrieval, 36th IEEE Conference on the Foundations of Computer Science (FOCS), pp. 41-50. (1995).

[5] Carlos, A., M., Chiffrement Homomorphe et Execution d'Algorithmes sur des Données Chiffrées: Avancées Récentes, PICC, XLIM. (2011).

[6] Zeeshan, P., Ammar, A. A., Asad, M. K., Sangyoung, L., Privacy-Aware Searching with Oblivious Term Matching for Cloud Storage, Springer J Supercomput 63:538–560. (2012).

[7] PFPDT :Préposé fédéral à la protection des données et à la transparence, Feldeggweg 1300 Berne. (2011).

[8] Kevin, H., Murat, K., Latifur, K., Bhavani, T., Security Issues for Cloud Computing, International Journal of Information Security and Privacy, 4 (2), 39-51. (2010).

[9] Carlos, A. M., Les Protocoles de Retrait d'Information Privé (RIP), XLIM UMR CNRS 6172, (2008).

[10] Adi, S., Identity-Based Cryptosystems and Signature Schemes. Advances in Cryptology, pp. 47-53. (1985).

[11] Zielinski, D., Be Clear on Cloud Computing Contracts. HRMagazine, Nov, 54(11), 63-65. (2009).

[12] Nashaat el-Khameesy, Hossam Abdel Rahman Mohamed. A Proposed Model for Datacenter in Depth Defense to Enhance Continual Security. International Journal of Information Technology and Computer Science, Vol 05, No. 04, 55-67, (2013).

[13] Talbot, D., Security in the Ether. Technology Review, 113(1), 36-42. (2010).

[14] Greengard, S., Kshetri, N., Cloud Computing and Developing Nations. Communications of the ACM, 53(5), 18-20. (2010).

[15] Xuyun, Z., Chang, L., Surya, N., Jinjun, C., An Efficient Quasi-Identifier Index Based Approach for Privacy Preservation over Incremental Data Sets on Cloud, Journal of Computer and System Sciences 79 542–555. (2012).

[16] Melvin, G., John, W. N., 2014, Cloud Computing Method and System. United State Patent No US8762709 B2, (2014).

[17] Cong, W., Ning, C., Kui, R., Wenjing, L., Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data, IEEE, ICDCS'10. (2010).

[18] Ming, L., Shucheng, Y., Wenjing L, Y., Thomas, H., Toward Privacy-Assured Cloud Data Services with Flexible Search Functionalities, SPCC-2012. (2012).

[19] Sion, R., Carbunar, B., On the computational practicality of private information retrieval. In NDSS 2007. (2007).

[20] Olumofin, F., Goldberg I., Revisiting the computational practicality of private information retrieval. In FC 2011. (2011).
[21] Dan, B., Matt, F., Identity-Based Encryption from the Weil Pairing. Advances in Cryptology | CRYPTOGRAPHIC 20012139/2001, pp. 213-229.doi: 10.1007/3-540-44647-8_13. (2001).
[22] Vipul, G., Amit, S., Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. Conference on Computer and Communications Security: Proceedings of the 13th ACM conference on Computer and communications security. (2006).
[23] Bethencourt, J., Sahai, A., Waters, B., Ciphertext-Policy Attribute-Based Encryption. Proceedings of 2007 IEEE Symposium on Security and Privacy. (2007).
[24] Jonathan, K., Amit, S., Brent, W., Predicate Encryption Sup-porting Disjunctions, Polynomial Equations, and Inner Products'. Advances in Cryptology EUROCRYPT2008, pp. 146-162. (2008).
[25] Dan, B., Brent, W., Conjunctive, Subset, and Range Queries on Encrypted Data. Theory of Cryptography. Ed. by Salil Vadhan. Vol. 4392. Lecture Notes in Computer Science. Springer Berlin/Heidel-berg, pp.535-554. (2007).

Authors' Profiles

Mahmoud Fahsi: PhD student at EEDIS laboratory, Djillali Liabes University, and also an Assistant at the Computer Science Department, Taher Moulay university, Algeria. Interested by information retrieval query reformulation and privacy research area.

Sidi Mohamed Benslimane: As an Associate professor, Dr. Benslimane is actually the head of the national computer science school, Sidi bel abbes, Algeria. Also he is the Head of Research Team 'Service Oriented Computing' at the EEDIS Lab. Stearing member of the ICWIT conference and program committee member of many international conferences around the world.

Amine Rahmani: PhD Post-graduated student for doctor degree at Taher Moulay university, Algeria and Member of the GECODE Lab. interested by information security and privacy for big data.

How to cite this paper: Mahmoud Fahsi, Sidi Mohamed Benslimane, Amine Rahmani,"A Framework for Homomorphic, Private Information Retrieval Protocols in the Cloud", IJMECS, vol.7, no.5, pp.16-23, 2015.DOI: 10.5815/ijmecs.2015.05.03

