International Journal of Security and Its Applications
Vol. 11, No. 12 (2017), pp.79-88
http://dx.doi.org/10.14257/ijsia.2017.11.12.07

Searchable Encryption: A Review

Khadijah Chamili[1, 2], Md. Jan Nordin[2], Waidah Ismail[3] and Abduljalil Radman[3]

[1]*Universiti Sains Islam Malaysia,*
[2]*Faculty of Information Science & Technology, Universiti Kebangsaan Malaysia*
[3]*Faculty of Science & Technology, Universiti Sains Islam Malaysia*
khadijah@usim.edu.my, jan@ukm.edu.my, waidah@usim.edu.my,
abdu_rad@yahoo.com

Abstract

Cloud computing is one of the most important technologies which supports reliability, scalability, ease of deployment and cost-efficient to business growth. Despite its benefits, cloud computing still has open and remain challenges on ensuring confidentiality, integrity, and availability (CIA) of sensitive data located on it. As a solution, the data is encrypted before sending to the cloud. However, the normal searching mechanism could not get through the encrypted data. In this paper, Searchable Encryption (SE) techniques which allow accessing data on encrypted cloud were reviewed. Nine SE techniques were presented with different issues and challenges on achieving secrecy and efficiency of SE. Four factors with their characteristics of SE were also identified for novice reader as a guidance of their future works.

Keywords*: Searchable Encryption, Cloud computing, Encryption, Factors, Review, Survey*

1. Introduction

Cloud computing is one of advent technologies that support easy deployment application systems with a low-cost implementation which offer Pay-Per-Use basis [3], [4], [5], [6], [7]. The emergence of cloud or mobile computing dramatically emerges the growth of the mobile commerce, mobile learning, mobile health and mobile gaming [8] which acquire availability and reliability of services 24 by 7. These make cloud computing services on demand and in trend due to its scalability, dynamic provisioning, ease of integration and support multi-tenant [8].

Confidentiality, integrity, and availability (CIA) of personal information (*e.g.* identity number, telephone number, address) are still remaining challenges faced in cloud computing [1]. Cloud computing is always considered as an untrusted or semi-trusted server since all the servers and storage are located physically at Cloud Service Provider (CSP) premise [3], [5]. This untrusted server tends to contribute to the privacy and security issues [3], [4], [5], [6]. One of the main security issues is data confidentiality. Data confidentiality is to ensure sensitive data (*e.g.* personal information) is safe from unauthorized access. Recently, cryptography or encryption is regarded as a promising method for data confidentiality [5], [7], [8].

Cryptography or encryption is about to secure a communication over an insurer channel [9]. However, traditional search mechanisms do not work for encrypted data [10], [11]. One of the solutions is using Searchable Encryption (SE) which allow users to search on encrypted data stored in the cloud in a secure manner [4]. In this paper, the searchable encryption methods developed in the literature are reviewed and classified based on technique utilized.

Received (August 7, 2017), Review Result (November 9, 2017), Accepted (November 24, 2017)

ISSN: 1738-9976 IJSIA

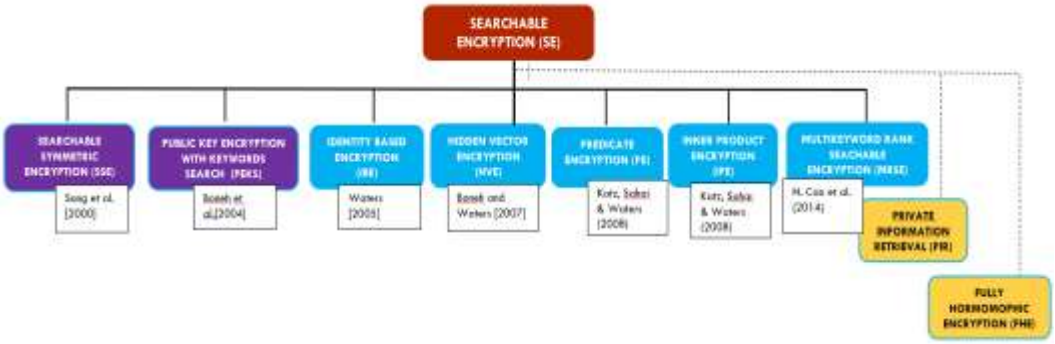

Figure 1. Searchable Encryption Techniques

The rest of this paper is organized as follows. We first review issues motivate to SE works and nine SE techniques with related works in Section II. In Section III, we discuss factors which affect SE performance. 4. Lastly, we conclude this paper with a suggestion on combination SE technique while applying SE in future works.

2. Searchable Encryption (Se)

SE is defined as searching for an encrypted data located on untrusted server or cloud without the server learn anything from the data [11]. In 2000, Song *et al*. proposed the idea of SE scheme which solves the issue on searching encrypted data on the cloud. According to C. Bosch *et al.* [12], SE scheme has six SE techniques (Fig. 1) and is still rapidly growing until now. These six SE techniques include Searchable Symmetric Encryption (SSE), Public Key with Keyword Search (PEKS), Identity-Based Encryption (IBE), Hidden Vector Encryption (HVE), Predicate Encryption (PE), and Inner Product Encryption (IPE). Moreover, a recent study [3] has regarded that Multi-keyword Rank Searchable Encryption (MRSE) as a new SE technique. Based on the literature review has been done for last seven years as shown in Fig. 2, we can see the intention from academia and industry in SE are tremendously increased from year to year.

However, the techniques are shown in Figure 1 remains unchallenged issues. They are developed to maintain secure and efficient communication between client and server on cloud [11]. They also support single user architecture [11] and multi-user architecture [10]. In addition, works done to support single keyword search [11], [13], multi-keyword search and ranking [3], [14], [15], subset query and range query [16], [17] and fuzzy multi-keyword search [18], [19]. Most of the previous work done on SE to improve secrecy motivated by adversary activity [20], [21]. Some adversary activities include brute force attack [20], search/access pattern leakage [8], [21] and DDoS attack [5]. To proof that any scheme or algorithm developed was resistant enough from attacks or adversary activities, system model and threat model were constructed for experimental purposes [3], [14]. In other works, other techniques like Private Information Retrieval (PIR) [22] and Fully Homomorphic Encryption (FHE) [23] were considered in enhancing SE.

2. 1. Symmetric Searchable Encryption

Symmetric Searchable Encryption (SSE) allows the user to upload data to the cloud with provable secrecy by issuing isolated and hidden query [11]. Hidden query and isolation query allow the server to learn nothing about the plaintext except the ciphertext. The query is running as an encrypted query which called as trapdoors: trapdoors are always generated using secret key [11]. The SSE probabilistic algorithm as below:

- Keygen(1^k): a key generation algorithm run by the data owner. It takes a security parameter k as input and outputs a secret key K.
- BuildIndex(K, D): a keyword index generation algorithm run by the data owner. It takes a secret key K and a set of documents D as inputs and outputs a keyword index I.
- Trapdoor(K, w): a keyword trapdoor generation algorithm run by the user. It takes a secret key K and a query keyword w as inputs, and outputs the trapdoor Tw for the keyword w.
- Search(I, Tw): a keyword search algorithm run by the server. It takes a keyword index I and a trapdoor Tw as inputs and outputs a set of documents D(w) that contains query keyword w. , SSE is considered more suitable for outsourcing data of company or organization application system on the cloud or untrusted server [4], [11]. It supports client/server architecture. For example, Alice encrypts data with her secret key which generate by KeyGen algorithm. To encrypt data, two algorithms run: Enc algorithm and BuildIndex algorithm [4].

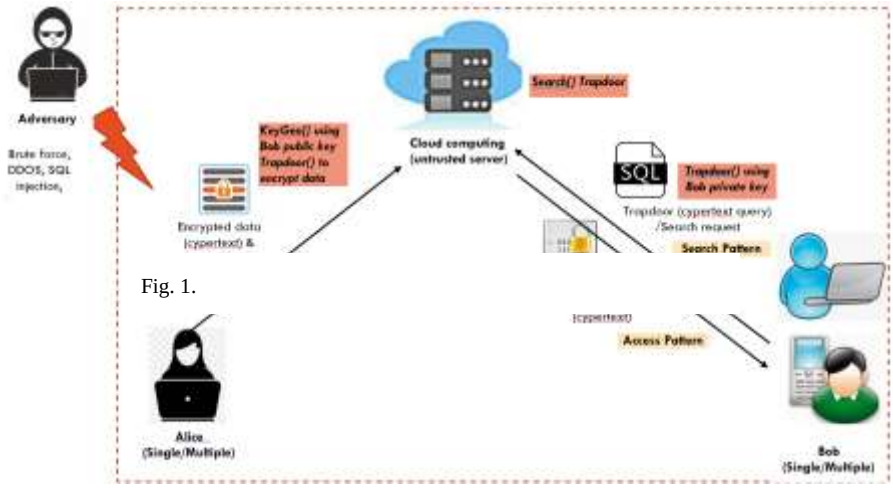

Fig. 1.

Figure 2. Framework of Searchable Data

These algorithms generate ciphertext (encrypted data) together with an encrypted index. Then, encrypted data and its encrypted index were sent to the cloud. To access the encrypted data, Bob will runs a query or trapdoor by issuing Trapdoor algorithm. Trapdoor algorithm encrypts the query request to the server on behalf of Bob. Next, the search algorithm computes the Trapdoor with encrypted index before the result is returned. The framework in Figure 3 illustrates SE.

SSE searching time was linear in the number of words in the message which consider as a practical solution. However, SSE face another security issue on statistical analysis (*e.g.* access pattern), one would run the same trapdoors for few times could expose to an adversary through statistic approaches.

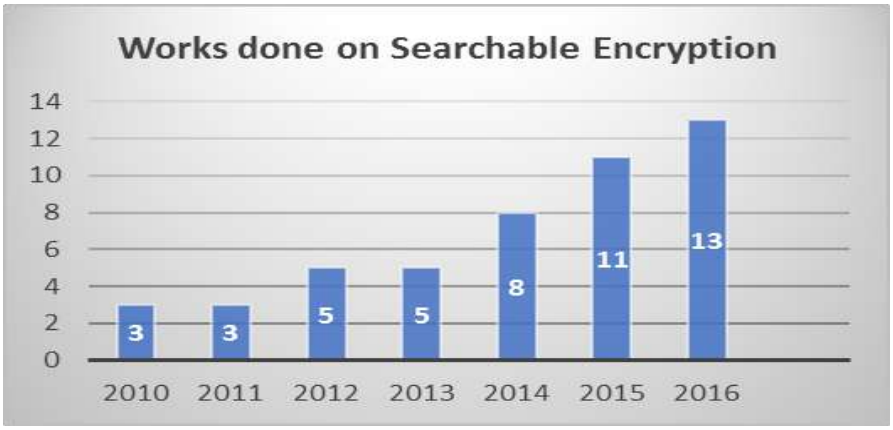

Figure 3. Research works on Searchable Encryption within 7 years

To overcome the security issue in Song et. al scheme, Goh [24] construct a secure IND-CKA index associated to each encrypted data file called Z-IDX which utilized Bloom filter. This secure index is semantically security against adaptively chosen keyword attack (IND-CKA) and a slightly stronger IND2-CKA. On the other hand, Curtmola *et al.* [25], revisited the SSE definition by using Oblivious RAM for stronger security definition which does not leak any information to the attacker/adversary.

In recent works, S. Dai *et al.* [26] constructed two memory leakage-resilient searchable symmetric encryption (MLR-SSE) scheme based on SSE and physic unclonable functions (PUFs) [27]. The PUFs is an equal function like hash function where it applies one-way function. The one-way function was introduced by O. Goldreich *et al.* [28] where it has below criteria :

- Easy to compute: There exists a deterministic P-time algorithm A such that on input x, A outputs f(x) (that is, A(x) = fix))
- Hard to invert: For every probabilistic P-time algorithm A', every positive polynomial p, and all sufficiently large n Pr(A'(f(U)).

The combination of these two schemes enables high protection for the user's private key and efficiency. Efficiency only achieves with generating the secret key in real time by PUFs.

Furthermore, SSE has been applied to support secure channel between multiple located data [8]. C. Liu *et al.* [8] developed Multi-Data-Source DSSE (MDS-DSSE) to support dynamic social data such as chatting application system which proven secured against adaptive chosen-keyword attacks (CKA2). MDS-DSSE work was based on Dynamic Searchable Symmetric Encryption (DSSE) [29], which is developed to support searching on encrypted data for very large datasets without leakage any information to the unauthorized use.

2.2. Public Key with Keyword Search

Public Key with Keyword Search (PEKS) was introduced by Boneh *et al.* [13] in 2004. Boneh *et al.* were using receiver's public-key for an encrypted message with keyword and the only receiver would allow decrypting using he/she private key. PEKS consist of 5 probabilistic polynomial-time algorithms as below:

- KeyGen(λ)→ (pkSE, skSE): Given a security parameter λ, the public/ private key pair (pkSE, skSE) is generated.
- Enc(pkSE,m)→c: Given the public key pkSE and a message m, it generates a ciphertext c.

- PEKS(pkSE,w)→Sw: Given the public key pkSE and a keyword w, it generates a PEKS ciphertext Sw of w.
- Trapdoor(skSE,w)→Tw: Given a keyword w and the private key skSE, it produces a trapdoor Tw.
- Test(pkSE,Sw,Tw')→{0, 1}: Given the public key pkSE, a searchable encryption ciphertext Sw, and a trapdoor Tw', it outputs 1 (true) if w = w' or 0 (false) otherwise. PEKS is mainly developed for data sharing scenario [4], [12], [13], for example putting mail on Email Service Provider like Gmail or Yahoo. With PEKS, Alice encrypts the data with Bob's public key by issuing KeyGen algorithm. Once data requested by Bob through Trapdoor algorithm (query is in an encrypted manner) using Bob's private key, Test algorithm will compare keyword search and return to Bob if success.

Z. Deng *et al.* [10] used asymmetric searchable encryption to design multi-user searchable encryption scheme with keyword authorization (MSESKA). To construct a multi-user setting, 6 polynomial-time algorithms were used. H.Yin *et al.* [17] developed a secure index technique using Decisional Diffie–Hellman (DDH) and Bilinear Diffie–Hellman (BDH) assumptions together with bloom filter technique in order to secure the search scheme. Chosen secret keys were generated randomly for every query trapdoors requested to the server. This ensures the efficiency and enhanced the secure search scheme.

In a different work, Secure Hybrid Indexed Search (SHIS) scheme was developed by W. Wang *et al.* [30] based on PEKS and DE. The main idea behind the SHIS is to reduce the search complexity on PEKS by applying Dynamic Index (DI) and Static Index (SI). SI is used for the first time search while DI is applied on the next query submission.

2.3. Identity-Based Encryption

Identity-Based Encryption (IBE) algorithm was initiated in 1984 by Shamir [31]. This scheme uses user's identity as a key for encryption and decryption process. User's identity key can be publicly accessed, which means that anyone can use it for sending a message. On the other hand, only the recipient with the private key can decrypt the message. For example, imagine the analogy of sending and receiving email, where recipient's email address which based on the recipient's name is used to send an email from the sender, while the recipient can open the email by using his/her email address.

PEKS is one of the main SE techniques that developed based on IBE [13]. IBE system which constructed by Bilinear Diffie-Hellman (BDH) proved that PEKS in the random oracle model is semantically secure against a chosen keyword attack. X. Dong *et al.* proposed a secure, efficient and scalable data collaboration scheme (SECO), in order to overcome multiple users setting encryption, allow write and updating query including fine-grained access control issues in cloud communication. These only happened by adopting two-level hierarchical identity-based encryption (HIBE) [32]. With SECO, data was encrypted with multiple recipient's public keys and only those users have the secret key would be allowed to access the data has been assigned to them. In this scheme, BDH was constructed in order to ensure SECO provides semantically secure and probabilistic.

2.4. Hidden Vector Encryption

Hidden Vector Encryption (HVE) is a type of predicate encryption (PE) that supports conjunctive equality and range of queries on encrypted data such as equality queries, comparison queries, and subset queries[33]. HVE is a specialized type of PE where ciphertext and token were associated with two vectors over attributes. At a higher level, the ciphertext matches the token if and only if the two vectors are component-wise equal[34].

2.5. Predicate Encryption

Predicate encryption (PE) allows users who use the public key on encrypted data without a private key. In a PE scheme, instead of using the full private key, the token is provided to a query server. The query server then performs a test to ensure token supplied were matched the ciphertext. If the test succeeds, the query server then forwards the encrypted data to the public key owner without revealing any information to the server [35].

V. Goyal *et al.* [36] introduced Attribute-Based Encryption (ABE) which allows the sender to define who should be able to read the data by setting up policy. In this scheme, an authority responsible to distribute the private keys together with sets of formulas over attributes and ciphertexts. A user with the distributed private key would able to decrypt the ciphertext as well as able to read the plaintext.

X.A. Wang *et al.* [37] claimed that PE could achieve more sophisticated and flexible functionality compared to the traditional public key encryption by transforming PE to PEFKS for example. This transformation would allow the scheme to support multiple keyword searches which supporting conjunctive or disjunctive logical rather than the equal relation only. According to J. Katz [35] IBE, Anonymous IBE (AIBE) and attribute-based encryption schemes support range queries which considered under PE's framework.

2.6. Inner Product Encryption (IPE)

Inner Product Encryption (IPE) was first introduced by J. Katz *et al.* [35] which known as a cryptographic mechanism that allows more fine-grained [33] (facilitate user with access to the data which fulfill the needs and requirement of the task given) with control over access to encrypt data. IPE cryptographic or known as *inner product computation* are most used in PE, IBE and HVE [38]. J. Katz *et al.* also managed to construct *attribute-hiding* schemes which handle disjunctions on polynomial-time predicates that are different from *payload-hiding* [33]. *Payload-hiding* is security notion to achieve stronger security level guarantees, where the ciphertext associated with an attribute that hides all information until the secret key is possessed to decrypt. *Payload* and *attribute-hiding* slightly different in a way of ciphertext conceal of the plaintext. For *attribute-hiding*, the associated parameter should be concealed together with the ciphertext while *payload-hiding* only requires the plaintext concealed together with the ciphertext [39].

2.7. Multi-keyword Ranked Search Encryption

Multi-keyword ranked search over encrypted cloud data (MRSE) was introduced in 2014 by N. Cao *et al.* [14]. The main idea of this scheme was to allow users on search request and return documents with semantic multiple keywords through "inner product similarity" keywords. In order to secure and get the most relevant results retrieval, MRSE was adapted from secure k-nearest neighbor (kNN) technique to select the k nearest database records between database record (p_i) and query vector (q). Secure inner product computation was adopted in order to set strict privacy requirement to ensure secrecy of cloud communication [14].

However, MRSE has three major drawbacks defined by R. Li *et al.* [3]. First, MRSE is using a static dictionary which needs the dictionary to rebuild for every additional keyword, result presented in out-of-order form which difficult for user to get the most relevant file and lastly, MRSE does not consider the weight of keyword and access frequencies where keyword's file is not in the top list of the result. Therefore, R. Li proposed new flexible multi-keyword query scheme called MKQE to overcome MRSE's drawbacks. MKQE have successfully solved the keyword dictionary expansion issue by implementing the partitioned matrices approach. Furthermore, MQKE uses the keyword's weight in the index file to solve the out-of-order problem in the matching result set.

2.8. Private Information Retrieval

Private Information Retrieval (PIR) protocol allows multiple readers to retrieve *i*th of *n*th bit data from multiple databases without revealing any information including access/search pattern to the server. PIR works significantly in smaller communication complexity than the obvious n-bit solution. It was first introduced in 1995 by E. Kushilevitz *et al*. [40]. Although data in PIR is always in a unencrypted manner while storing at the server, PIR has been referred by the most researcher in SE work because of its features to support secrecy and efficiency of SE.

Fully Homomorphic Encryption

FHE scheme is another technique to ensure the ciphertext size shorten and the complexity of decryption reduced through re-linearization (*i.e.* a process in reducing the size of the ciphertext back down to n+1) [41]. According to Gentry, FHE scheme security is strong enough and semantically secure [42]. Z. Brakerski [41] has applied FHE in SE by converting the symmetric ciphertexts into homomorphic ciphertexts without additional communication. X. Yi *et al.* [43] had developed single-database with PIR protocol using FHE which allows data to be encrypted only bit by bit in block database. This contributes to communication complexity $O(\underline{y}\log m+\underline{y}n/m)$ higher than $O(\log^2 n)$. Using PIR, communication is strictly smaller than n.

In the other hand, L. Tajan *et al.*[22] combined PIR protocol and Somewhat Homomorphic Encryption (SHE) together with SE in order to hide the data store evidence which affected by the computation.

3. Discussion

Based on our review, we found that SSE and PEKS are the most popular SE techniques used among the rest SE techniques described in Section II. In general, SE technique is not limited to search data but can be extended to add, delete and edit the data in the cloud. SE technique should be applied accordingly to the needs of the application system: secrecy, efficiency, architecture and keyword search. The factors of SE technique and its characteristics are tabled in Table 1.

Below are the characteristics for each of SE factor:

- *Encryption/ Decryption Complexity*: SSE is using symmetric encryption, hence it would be combined with other scheme or technique to build complex encryption key. Work by S. Dai et.al. is one of the examples. On the other side, PEKS should be considered as complex encryption key where it applied asymmetric key.
- *Attack*: “indexes known as semantic security against adaptively chosen keyword attack” (INDK-CKA) has been formulated by Goh as a security model of SE towards chosen keyword attack. This security model was used to prove scheme developed under SE was semantically secured. Furthermore, the scheme should also be considered constructed securely in the random oracle model under Decisional Diffie–Hellman (DDH) or Bilinear Diffie–Hellman (BDH) assumptions.
- *Access Control*: IBE, HVE, PE and IPE technique are fine-grained access control technique which could be applied to support role-based function. By using access control technique, will limit user access to the data being defined for them.
- *Secret Key Size*: The length of the key is closely related to the complexity of encryption and decryption. FHE should be considered in developing new SE scheme since it able to shorten the ciphertext and reduce the complexity of encryption and decryption to enhance the efficiency of SE.
- *Index Size*: Using indexing will fasten the searching speed of ciphertext. However, the length of the index could impact the Query Time.

- *Query Time*: The efficiency of SE is strictly influenced by the query or searching time. Each data was associated with an index where index should be organized accordingly to enhance the efficiency. Tree structure, Static Index (SI) and Dynamic Index (DI) and Bloom Filter Array could be considered.
- *User Setting*: How user would access the data is the another characteristic need to be considered. Single user vs multi-user setting, mobile user vs desktop user would impact the efficiency and secrecy factor. Z. Deng *et al.* studied how SE could support for multi-user architecture.
- *Database Setting*: Database were located at a different location to support search engine architecture *e.g.* Google, Yahoo. At this point of setting, security issue would raise and most suitable SE technique recommended is PEKS with fine-grained access control implemented.
- *Single Keyword*: Most earlier works in SE scheme such as SSE and PEKS were supported single keyword searching. Using XOR function limit the searching in order to tolerate minor typos while searching.
- *Multiple Keyword*: "inner product similarity" is one of the methods applied to support multiple keywords.
- *Boolean Keyword*: HVE and IPE would suitable to use to achieve Boolean keyword where query support conjunctive combinations of equality, comparison, and subset predicates.
- *Fuzzy Search*: Due to the limitation on keyword searching *e.g.* search 'netwrk' instead of 'network', fuzzy search tolerate with minimum typos. Unigram or bigram, Bloom Filter with Storage (BFS) or Locally Sensitive Hashing (LSH) would be considered to achieve Fuzzy Search.

However, for best implementation on ensuring privacy and efficiency, few techniques should be combined. This helps in constructing semantic security and boost efficiency on searching.

For a further understanding of SE, we may refer a sample of library system where data collection is kept in a server with metadata (tag/index) for searching or crawling purposes. In SE, the way data is kept in the server by applying metadata (tag/index) is almost the same but in encrypted mode.

Table 1. Factors of Searchable Encryption

Factor	Characteristic
Secrecy	Encryption/ Decryption Complexity [3],[8],[10],[26] Attack [5],[8],[20],[21] Access Control [3],[22]
Efficiency	Secret Key Size [3],[16],[26] Index Size [10],[19] Query Time [8],[19]
Architecture	User Setting [10], [22],[32] Database Setting [8], [29]
Keyword Search	Single Keyword [11], [13] Multiple Keyword [3], [14], [15] Boolean Keyword [16], [17] Fuzzy Search [18], [19]

In this paper, we only discuss the SE techniques without the schemes for each SE techniques. As we go in deep of this domain, we understand so many schemes formulated for each of the SE techniques. However, the scope of this initial work only covers the SE technique which we think good enough to provide an overview of SE for novice reader for their further reading.

4. Conclusion

We present nine SE techniques: SSE, PEKS, IBE, PE, IPE, HVE, MRSE, PIR and FHE as SE's family. SE techniques allow the user search on the encrypted cloud. The main goal of these SE techniques is to build secure and efficient communication between user and cloud. Four (4) main factors were identified which affect the SE performance on the cloud: efficiency, secrecy, architecture and keyword search. With the aim to list all SE techniques from the year 2000 until recent years and factors influence SE performance, we hope to help novice readers to understand SE for their further reading. In conclusion, to develop proven and semantically secure scheme, the combination of SE techniques should be considered in future works.

Acknowledgment

This project is funded by Newton-Ungku Omar Fund: GRANT USIM/INT-NEWTON/FST/IHRAM/053000/41616.

References

[1] J. L. Fernández-Alemán, I. C. Señor, P. ángel O. Lozoya and A. Toval, "Security and privacy in electronic health records: A systematic literature review", *J. Biomed. Inform.*, vol. 46, no. 3, **(2013)**, pp. 541–562.
[2] Y. Yang, X. Zheng, and C. Tang, "Lightweight distributed secure data management system for health internet of things", *J. Netw. Comput. Appl.*, in press, **(2016)**.
[3] R. Li, Z. Xu, W. Kang, K. C. Yow and C. Z. Xu, "Efficient multi-keyword ranked query over encrypted data in cloud computing", *Futur. Gener. Comput. Syst.*, vol. 30, no. 1, **(2014)**, pp. 179–190.
[4] F. Han, J. Qin and J. Hu, "Secure searches in the cloud: A survey", *Futur. Gener. Comput. Syst.*, in pres, **(2015)**s.
[5] K.-K. R. Choo, J. Domingo-Ferrer and L. Zhang, "Cloud Cryptography: Theory, Practice and Future Research Directions", *Futur. Gener. Comput. Syst.*, vol. 62, **(2016)**, pp. 51–53.
[6] Q. Liu, A. Srinivasan, J. Hu and G. Wang, '"Preface: Security and privacy in big data clouds"', *Futur. Gener. Comput. Syst.*, vol. 72, **(2017)**, pp. 206–207.
[7] S. K. Pasupuleti, S. Ramalingam and R. Buyya, '"An efficient and secure privacy-preserving approach for outsourced data of resource constrained mobile devices in cloud computing"', *J. Netw. Comput. Appl.*, vol. 64, **(2016)**, pp. 12–22.
[8] C. Liu, L. Zhu and J. Chen, '"Efficient searchable symmetric encryption for storing multiple source data on cloud"', *Proc. - 14th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. Trust. 2015*, in press, vol. 1, **(2015)** , pp. 451–458.
[9] S. Goldwasser, "Lecture Notes on Cryptography", unpublished, no. July, **(2008)** , pp. 1–289.
[10] Z. Deng, K. Li, K. Li and J. Zhou, "A multi-user searchable encryption scheme with keyword authorization in a cloud storage", *Futur. Gener. Comput. Syst.*, in press, **(2016)**.
[11] D. X. Song, D. Wagner and A. Perrig, "Practical techniques for searches on encrypted data", *Proceeding 2000 IEEE Symp. Secur. Priv.*, **(2000)**, pp. 44–55.
[12] C. Bösch, P. Hartel, W. Jonker and A. Peter, "A Survey of Provably Secure Searchable Encryption", *ACM Comput. Surv.*, vol. 47, no. 2, **(2014)**, pp. 1–51.
[13] D. Boneh, G. Di Crescenzo, R. Ostrovsky and G. Persiano, "Public Key Encryption with Keyword Search", *Proc. 23rd Annu. Int. Conf. Theory Appl. Cryptogr. Tech.*, **(2004)**, pp. 506–522.
[14] N. Cao, C. Wang, M. Li, K. Ren and W. Lou, "Privacy-Preserving Multi-keyword Ranked Search over Encrypted Cloud Data", no. 1, **(2014)**.
[15] Y. Liu, Z. Li, W. Guo and W. Chaoxia, "Privacy-preserving multi-keyword ranked search over encrypted big data", *Third Int. Conf. Cybersp. Technol. (CCT 2015)*, no. 1, **(2015)**, pp. 1–3.
[16] B. Zhang and F. Zhang, "An efficient public key encryption with conjunctive-subset keywords search", *J. Netw. Comput. Appl.*, vol. 34, no. 1, **(2011)**, pp. 262–267.
[17] H. Yin, Z. Qin, L. Ou and K. Li, "A query privacy-enhanced and secure search scheme over encrypted data in cloud computing", *J. Comput. Syst. Sci.*, vol. 11, no. 16, **(2016)**, pp. 311–14.

[18] M. Chuah and W. Hu, "Privacy-aware bedtree based solution for fuzzy multi-keyword search over encrypted data", *Proc. - Int. Conf. Distrib. Comput. Syst.* , **(2011)**, pp. 273–281.
[19] Z. Fu, X. Wu, C. Guan and X. Sun, "Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement", *IEEE Trans.*, no. July, **(2016)**.
[20] L. Fang, W. Susilo, C. Ge and J. Wang, "Public key encryption with keyword search secure against keyword guessing attacks without random oracle", *Inf. Sci. (Ny).*, vol. 238, **(2013)**, pp. 221–241.
[21] M. S. Islam, M. Kuzu and M. Kantarcioglu, "Access Pattern disclosure on Searchable Encryption: Ramification, Attack and Mitigation", *Ndss'12*, **(2012)**.
[22] L. Tajan and C. A. Reuter, "Private Information Retrieval and Searchable Encryption for Privacy-Preserving Multi-Client Cloud Auditing", pp. 162–169, **(2016)**.
[23] A. a Atayero and O. Feyisetan, "Security Issues in Cloud Computing : The Potentials of Homomorphic Encryption", *J. Emerg. Trends Comput. Inf. Sci.*, vol. 2, no. 10, **(2011)**, pp. 546–552.
[24] E.-J. Goh, "Secure Indexes", *An early version this Pap. first Appear. Cryptol. ePrint Arch. Oct. 7th*, **(2003)**, pp. 1–18.
[25] R. Curtmola, J. Garay, S. Kamara and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions", *In Proceedings of the 2006 ACM Conference on Computer and Communications Security, CCS'06*, **(2006)**, pp. 79–88.
[26] S. Dai, H. Li and F. Zhang, Memory leakage-resilient searchable symmetric encryption', *Futur. Gener. Comput. Syst.*, vol. 62, pp. 76–84, 2016, in press.
[27] P. S. Ravikanth, 'Physical One-Way Functions', *Science (80-.).*, 2002, unpublished.
[28] O. Goldreich, L. a. Levin, and L. a Levint, "A hard-core predicate for all one-way functions", *Proc. twenty-first Annu. ACM Symp. Theory Comput. - STOC '89*, **(1989)**, pp. 25–32.
[29] D. Cash, J. Jaeger, S. Jarecki, C. Jutla, H. Krawczyk, MC. Rosu and M. Steiner, "Dynamic Searchable Encryption in Very-Large Databases: Data Structures and Implementation", *Proc. 2014 Netw. Distrib. Syst. Secur. Symp.*, no. February, **(2014)**, pp. 23–26.
[30] W. Wang, P. Xu, H. Li and L. T. Yang, "Secure hybrid-indexed search for high efficiency over keyword searchable ciphertexts", *Futur. Gener. Comput. Syst.*, vol. 55, **(2016)**, pp. 353–361.
[31] '1998 (Shamir) - ID-basedCryptoSystem.pdf'. .
[32] X. Dong, J. Yu, Y. Zhu, Y. Chen, Y. Luo and M. Li, "SECO: Secure and scalable data collaboration services in cloud computing", *Comput. Secur.*, vol. 50, **(2015)**, pp. 91–105, in press.
[33] J. Katz, A. Sahai and B. Waters, "Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products", pp. 146–162.
[34] J. H. Park, "Efficient hidden vector encryption for conjunctive queries on encrypted data", *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 10, **(2011)**, pp. 1483–1497.
[35] J. Katz, A. Sahai and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products", *J. Cryptol.*, vol. 26, no. 2, **(2013)**, pp. 191–224.
[36] V. Goyal, O. Pandey, A. Sahai and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data", *Proc. 13th ACM Conf. Comput. Commun. Secur. - CCS '06* , **(2006)**, p. 89.
[37] X. A. Wang, F. Xhafa, W. Cai, J. Ma and F. Wei, "Efficient privacy preserving predicate encryption with fine-grained searchable capability for Cloud storage", *Comput. Electr. Eng.*, vol. 0, **(2015)**, pp. 1–13.
[38] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima and B. Waters, "Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption", vol. 2, no. subaward 641, pp. 62–91.
[39] T. Okamoto and K. Takashima, "Adaptively attribute-hiding (hierarchical) inner product encryption", *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E99A, no. 1, **(2016)**, pp. 92–117.
[40] B. Chor, O. Goldreich, E. Kushilevitz and M. Sudan, "Private information retrieval", *J. ACM*, vol. 45, no. 6, **(1998)**, pp. 965–982.
[41] Z. Brakerski, "Efficient Fully Homomorphic Encryption from (Standard) LWE", **(2011)**, pp. 97–106.
[42] C. Gentry, "Computing arbitrary functions of encrypted data", *Commun. ACM*, vol. 53, no. 3, **(2010)**, p. 97.
[43] X. Yi, M. G. Kaosar, R. Paulet and E. Bertino, "Single-database private information retrieval from fully homomorphic encryption", *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 5, **(2013)**, pp. 1125–1134.

