International Journal of Technology (2017) 4: 591-600
ISSN 2086-9614

USER VERIFIABLE MULTIPLE KEYWORD SEARCH SCHEME USING THE MERKLE TREE FOR OUTSOURCED DATA IN THE CLOUD

Devi Thiyagarajan[1*], R. Ganesan[1]

[1]*School of Computing Sciences and Engineering, VIT University, Chennai 600127, India*

(Received: November 2015 / Revised: February 2017 / Accepted: July 2017)

ABSTRACT

Cloud computing has revolutionized the IT industry by offering huge storage for data outsourcing and also for computation. Various security issues concerned with security and privacy of data arise in the context of data outsourcing. The framework enables clients to outsource encrypted data to the cloud, enables users to retrieve preferred documents using multiple keywords and allows the user to verify the response from the server. The strength of the proposed model relies on the discrete logarithmic problem of Hyper Elliptic Curve Cryptography (HECC) and the security of Merkle trees. The paper proposes a user verifiable multi-keyword search scheme, which focuses on: (i) construction of inverted index for the documents with keywords; (ii) index and document encryption by HECC; (iii) index and document authentication by the Merkle tree; and (iv) verification of the accuracy of response from server by top hash or root hash value of the Merkle tree. Security analysis and results demonstrate the correctness of proposed multiple keyword search (MKS) scheme. The search algorithm combined with the hash value verification process by the Merkle tree is strong enough to provide data security, privacy, and integrity. The proposed model reduces the storage overhead on both the client's and user's side. As the number of documents increases, the retrieval time is less, which reduces the storage overhead on both sides.

Keywords: Client; Cloud; Merkle tree; Search; Verifiable

1. INTRODUCTION

The cloud offers resources, such as storage and network, as services to customers and organizations. Few important characteristics of the cloud include on-demand self-service, broad network access, resource pooling, rapid elasticity and measured services (Figure 1). Deployment models of the cloud are public, private, community and hybrid clouds. Major service delivery models for the cloud are SaaS (Software-as-a-Service), PaaS (Platform-as-a-Service) and IaaS (Infrastructure-as-a-Service).

Through cloud computing, data outsourcing is the major advantage by which clients store their data on cloud servers. Upon retrieval of such stored data problems arise related to data security (Subhashini et al., 2011, Lee et al., 2013) Data encryption on the client side ensures that only encrypted data is stored on cloud servers which may prevent servers from misusing those data. Client side encryption provides full control of data to clients and thereby avoids any misbehavior of the server over encrypted data.

[*] Corresponding author's email: devi.janu@gmail.com, Tel. +91-416-2202811, Fax. +91-416-2202811

Permalink/DOI: https://doi.org/10.14716/ijtech.v8i4.9475

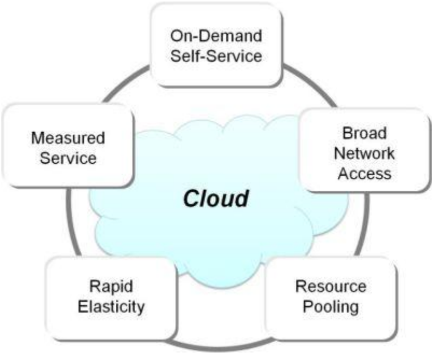

Figure 1 Characteristics of the Cloud

Searches over encrypted data with keywords (Cao et al., 2014) enables users registered with clients to retrieve the necessary file without the intervention of the client or verification by a third-party. Verifiable computation plays a major role in identifying the fraudulent activities of any third-party users on the server. Mostly the task of verifying the integrity of data stored in the cloud is done by a third-party user which may yield invalid results. Customers perform verification on the returned responses from cloud servers in order to maintain control over data. In cloud computing, one of the best data mining techniques, namely keyword searches, has been utilized to enhance user verifiable computation. The scenario discussed permits the data owner to outsource information to the cloud and at the same time it offers two major functionalities to users, such as: (i) use of multiple keywords to retrieve encrypted documents from the cloud; and (ii) correctness verification of the results returned from the server. User verifiability allows authenticated users to perform multiple keyword searches over documents on the server and at the same time verify the response from the server.

The building of an inverted index for document storage in the cloud and usage of the Merkle tree to verify the results of the server are important features of the proposed scheme. Comparison of the hash values of the position of keywords in documents with the encrypted document hash value, allows the user to decide on downloading the required data. The paper is organized as follows: The problem statement is defined in Section 2. Related work is discussed in Section 3. Section 4 proposes user verifiable multiple keyword search schemes. Section 5 discusses security analysis and Section 6 concludes this paper.

2. METHODOLOGY & PROBLEM STATEMENT

The client outsources the document collection ‘D’ to the cloud server ‘CS’ and provides the multiple keyword search function to authenticated users who are registered with the client. The client generates the public key Pk_D and the search key Sk_D. These keys together make the process of the user verifiable multiple keyword searches much easier. A registered user with the public key Pk_D can search the necessary documents and can verify the correctness of responses from the server. On receiving the query from a user, the server makes use of the search key Sk_D to provide proofs. The server can distinguish the verification key V_{key} from the public key Pk_D in order to check the correctness of the search result.

3. RELATED WORK

Proposed by Song et al. (2000), the scheme makes the user search the whole document for a specific single keyword. Newer schemes were proposed focusing on a single keyword search (Chang et al., 2005; Curtmola et al., 2006; Goh et al., 2003). Another search scheme, employing an inverted index was proposed by Curtmola et al. (2005), which provides an efficient search process, but keyword privacy is exploited. Rank order is secured by an order preserving method (Swaminathan et al., 2007; Wang et al., 2012; Zerr et al., 2009) where the search scheme is based on the term frequency and an inverted index. Symmetric searchable encryption along with implementation results is discussed by Salam et al. (2015). Sharable ID-based encryption employing a single keyword search scheme is also proposed by Xu et al. (2015). A public key cryptography (PKC)-based search scheme was proposed by Boneh et al. (2004), which allows the use of a public key for storing data and a private key for the search process.

Search over encrypted data is performed by a technique known as predicate encryption (Attrapadung et al., 2010; Shen et al., 2009; Shi et al., 2007). The conjunctive search scheme proposed by Hwang et al. (2007) also provides efficient document retrieval. A search scheme based on a vector space model, which experiences overhead on the user's side was proposed by Pang et al. (2010). This kind of scheme was unsuitable for the cloud environment because of an absence of security analysis. The privacy-preserving multi-keyword ranked search scheme proposed by Cao et al. (2011) provides a result for the search request by making use of coordinate matching. Lack of accuracy and the need for the server to traverse every index for every search request are noteworthy disadvantages.

Usage of symmetric searchable encryption and the absence of a user verifiability feature in the schemes proposed by Chai et al. (2012) and Kissel et al. (2013) make the schemes unsuitable for a successful search. The verifiable conjunctive keyword search scheme proposed by Cheng et al. (2015) also makes use of symmetric searchable encryption. Zheng et al. (2014) suggested Verifiable Attribute-Based Keyword Search (VABKS) which enables only user-satisfying access policies to retrieve documents from the cloud. Cao et al. (2014) proposed replication schemes, such as active index replication, proactive pointer replication, and passive index replication to improve query load imbalance.

Yang et al. (2015) proposed a hybrid technique for preserving the privacy of medical data. The 128-bit Advanced Encryption Standard (AES) algorithm for document encryption and Provable Data Possession for integrity-checking of remote data have been employed. The literature review about keyword search identifies the gaps that need to be addressed, such as: (i) position-based multiple keyword searches over encrypted data; and (ii) user verifiable search schemes with Merkle trees. The proposed schemes offer user verifiability which most of the search algorithms fail to provide.

4. USER VERIFIABLE MULTIPLE KEYWORD SEARCH

The client encrypts documents before uploading them to the cloud server by employing Hyper Elliptic Curve Cryptography (HECC). This multiple keyword search algorithm is proposed for efficient document retrieval from the cloud (Figure 2) and the Merkle tree technique is utilized to verify the integrity of the documents returned from the cloud on user request. The user verifiable MKS enables the client 'C' to outsource the set of documents 'D' to the cloud server 'CS,' while ensuring the following:

User verifiability: The user assesses the correctness of results sent back by the server CS. The user can verify whether the server has returned the correct document for the set of keywords K_s corresponding to MKS(D, K_s).

The user verifiable Multiple Keyword Search (MKS) can be defined by the following algorithms:

1. KeyGen (1^k, D) $\rightarrow$(Pk_D, Sk_D): The client runs KeyGen algorithm while outsourcing the set of documents D = {D_1, D_2, ……}. With input as λ security parameter, the algorithm generates the public key Pk_D as well as the search key Sk_D.

2. GenQuery (K_s, Pk_D) $\rightarrow$($EMKS_Q$, V_{key}): The user executes GenQuery algorithm when keywords K_s = {KW_1, KW_2, ….} is set and the public key Pk_D is given. The output of the algorithm is encoded as a multiple keyword search query $EMKS_Q$ and as the verification key V_{key}.

3. Search (Sk_D, $EMKS_Q$) $\rightarrow$$EMKS_R$: The server, when given the search key Sk_D and the encoded multiple keyword search query $EMKS_Q$, produces the encoding $EMKS_R$ of the search result D_{Ks}= MKS (D, K_s)

4. Verify ($EMKS_R$, V_{key}) $\rightarrow$ γ: Deterministic algorithms run by the user to check the integrity of result are returned by the server. The algorithm converts $EMKS_R$ to the search result D_{Ks}, which utilizes the verification key V_{key} to determine D_{Ks}= MKS (D, K_s). If the algorithm finds $\gamma = D_{Ks}$ and if D_{Ks} = MKS (D, K_s), then the user accepts the server result. If $\gamma = \perp$, then the user rejects the result.

Correctness: The user verifiable multiple keyword search is said to be correct, if when the cloud server CS invokes the 'Search' algorithm on the encoded multiple keyword search query $EMKS_Q$, then it produces the encoding $EMKS_R$, which the user 'U' always accepts.

Definition 1: The user verifiable multiple keyword search is correct, if for any set of documents 'D' and keyword collection K_s occurs as follows:

If KeyGen (1_k, *D*) $\rightarrow$(Pk_D, Sk_D), *GenQuery* (K_s, Pk_D) $\rightarrow$($EMKS_Q$, V_{key}) *and*

Search (Sk_D, $EMKS_Q$) $\rightarrow$ $EMKS_R$, *then:*

Pr (*Verify* ($EMKS_R$, V_{key}) $\rightarrow$*MKS* (*D*, K_s)) *=1*

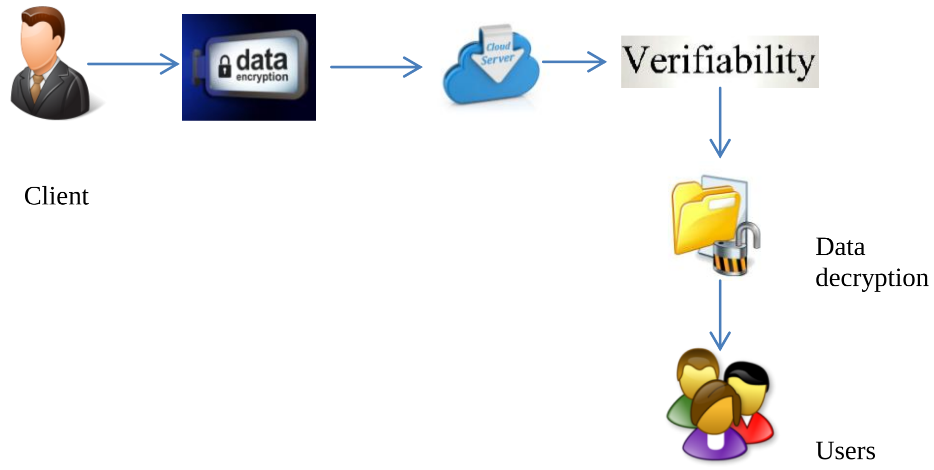

Figure 2 User verifiable search

4.1. Building Blocks

4.1.1. Merkle hash tree

The Merkle tree proposed by Merkle (1988) is used to compute the hash dynamically over encrypted data on the cloud. The user can verify whether the requested element 'd' is in the set X= {d_1, d_2, ….} by using Merkle trees. The following algorithm builds the Merkle tree for set X to authenticate elements in X.

1. M $\leftarrow$ BuildMHT (X, H): Algorithm builds Merkle tree T where each leaf in the tree is being mapped with each element d_i from the set X and hash of combining the children of a node is stored in every internal node.

2. path $\leftarrow$ GenerateMHTProof (T, d): The algorithm provides the authentication path for element d_i(leaf) which is actually siblings of nodes on the path from leaf to root γ.

3. {Accept, Reject} ← VerifyMHTProof (d, path, γ): The algorithm compares the root value calculated from d with the path to find whether they are equal to γ.

4.2. Scheme Description

The user verifiable multiple keyword search schemes enable the client to outsource document collection D to the cloud server after encrypting it as ED_i. The User 'U' can search in the encrypted data with the keyword set K_s and the response of the server is also verified by the User 'U.' The two major stages in the scheme are: (i) upload; and (ii) verifiable multiple keyword searches.

4.2.1. Upload

At the stage of the upload, the client runs the KeyGen algorithm with the λ security parameter and the document set 'D,' which generates the public key Pk_D and the search key Sk_D as the output. The following steps are carried out in the KeyGen algorithm:

1. The algorithm builds an inverted index Inv_Ind for the set of keywords K_s= {KW_1, KW_2, …., KW_{mn}}. The keyword set K_s contains the distinct keywords of the document set D.
2. The algorithm makes use of the Merkle tree TI to authenticate the inverted index Inv_Ind, where each leaf is mapped to a position of the document in Inv_Ind.
3. The algorithm constructs the Merkle tree TD with the hash of encrypted documents ED_i. The search key Sk_D= (EI, TI, TD, $\{ED_i\}_{1\leq i\leq n}$, K_s) is sent to the server.

```
•   $(Pk_D, Sk_D) \leftarrow KeyGen\ (1^k, D)$
    $D = \{D_1, D_2, \ldots\ldots, D_n\}$              // set of documents
    $K_s = \{KW_1, KW_2, \ldots., KW_{mn}\}$        // set of multiple keywords
Generation of the security parameter
H: $\{0,1\}^* \rightarrow F_p$ as function of the security parameter $\lambda$
Index construction and authentication    // creates an inverted index of size<D
Identify $KW_i$ from D
For $KW_i$ Є D
    BuildInv_Ind ($pos_i$, $K_s$, Doc_ID, D);
    Update Inv_Ind;
End
EI ← E (Inv_Ind)                           // Inverted index encryption
$ED_i$ ← E (D)                               // Document encryption
For $pos_i$ Є $D_i$ do
    Compute $PD_i$= $pos_i$ [$KW_i$ ($D_i$)];
    Compute $HP_i$= H ($PD_i$ || i)          // i is position of the document in the index
    TI= BuildMHT ($\{HP_i\}_{1\leq i\leq n}$, H)
End
For $ED_i$ Є Inv_Ind do
    Compute $HE_i$ = H ($ED_i$ || $KW_i$);    // top hash of encrypted document
    TD= BuildMHT ($\{HE_i\}_{1\leq i\leq n}$);
```

```
End
$CS \leftarrow EI, ED_i, HE_i$                    // encrypted content sent to cloud server
Return $Pk_D = (H, \delta_I, \delta_D)$;
Return $Sk_D = (EI, TI, TD, \{ED_i\}_{1 \le i \le n}, K_s)$
```

4.2.2. Verifiable multiple keyword search

The user needs to find the documents with the desired keywords in order that D_{Ks} is a subset of D. The algorithm GenQuery is run by the user which then outputs the search query $EMKS_Q = K_s$ and the verification key $V_{key} = (K_s, Pk_D)$. This query identified as $EMKS_Q$ is sent to the cloud server from the user.

After receiving the query $EMKS_Q$, the cloud server runs the search algorithm. The algorithm searches Inv_Ind for keywords KW_i Є K_s and determines the position of keywords in the documents. After finding the positions of keyword KW_i in the encrypted document ED_i, then the algorithm returns the correct decrypted original documents D_{Ks}. In order to identify that only valid documents are retrieved, the search authenticates positions of encrypted documents using the Merkle tree TI. The search also authenticates the encrypted documents by the Merkle tree TD.

The user checks the response sent from the server by calling up the command 'Verify algorithm.' The algorithm checks whether the returned positions pos_i of the documents D_{Ks} is correct using the path P_1, which is computed by the verification algorithm of the Merkle tree TI and also it also checks whether the returned documents are correct by using the path P_2, which is computed by the verification algorithm of the Merkle tree TD.

```
•   $(EMKS_Q, V_{key}) \rightarrow GenQuery\ (K_s, Pk_D)$
    Allot $EMKS_Q = K_s$ and $V_{key} = (K_s, Pk_D)$
    Return $\{EMKS_Q, V_{key}\}$
•   $EMKS_R \rightarrow Search\ (Sk_D, EMKS_Q, Inv\_Ind)$
For $KW_i$ Є $K_s$
    Determine $pos_i$ of $KW_i$ in EI;
    Determine $ED_i$ in EI;        // set of documents with preferred keyword in inverted index
    Update Inv_Ind;
    Compute path $P_1$= GenerateMHTProof (TI, $HP_i$);
                                   // compute path using positions and encrypted documents
    Compute path $P_2$= GenerateMHTProof (TD, $HE_i$);
    Return $EMKS_R = (D_{Ks}$, path $P_i)$
End
•   $(HP_i, HE_i) \rightarrow Verify\ (EMKS_R, V_{key})$
Analyze $V_{key} = (K_s, Pk_D)$
If $KW_i$ found in D then
    Process $EMKS_R = (D_{Ks}, \{path\ P_i\}_{1 \le i \le k})$
For $KW_i$ Є $K_s$
```

```
        If VerifyMHTProof ((HP_i||i), path P_1, γ_i) = Accept
 else
                return γ_i = Reject
 If VerifyMHTProof ((HE_i||i), path P_2, γ_D) = Accept
        else
                return γ_D = Reject
End
```

5. SECURITY ANALYS

The proof of the correctness of the proposed user verifiable multiple keyword search is discussed.

Correctness

Theorem 1:
The proposed search scheme is a correct user verifiable multiple keyword search.

Proof: A query $EMKS_Q = K_s= \{KW_1, KW_2, …., KW_{mn}\}$ is sent to the cloud server 'CS' by the User 'U.' The 'CS' runs and the search algorithm and sends back the response $EMKS_R$.

All keywords in K_s are found in D_{Ks}:

Then, $EMKS_R = (D_{Ks}$, path $P_i)$ where

D_{Ks} = set of documents with specified multiple keywords.

P_i = authentication path computed by the Merkle trees TI and TD, respectively.

The User 'U' accepts the decrypted documents from the 'CS,' if we presume the authentication of the positions of the documents and the encrypted documents by the Merkle tree to be correct.

The proof is generated by the Merkle tree as paths P_1 and P_2 help to identify whether the given document containing specific keyword D_{Ks} is returned to the User 'U.' Based on the proof, the user can accept or reject the content from the 'CS.'

5.1. Implementation Results

The OpenStack cloud is employed for implementation purposes, because of its features, such as scalability and elasticity. With a single keyword search, the time taken for retrieval of documents is slow compared to the multiple keyword search scheme in cloud computing (Figure 3). The X-axis denotes the entire number of the documents retrieved and the Y-axis denotes the total time taken for efficient retrieval of the documents from the cloud server by employing single and multiple keyword search schemes. The single keyword search scheme is an easily apparent scheme, but employing multiple keywords for the search reduces the search time practicality. This proves the efficiency of employing multiple keyword search schemes in the document retrieval process. The search algorithm combined with hash value verification process by the Merkle tree is strong enough to provide data security, privacy, and integrity. The proposed model reduces the storage overhead on both the client's and user's side (Figure 4). The X-axis denotes the percentage of documents retrieved and Y-axis denotes the number of documents on the client side. As the number of documents increases, the retrieval time is less, which reduces the storage overhead on the client's and user's side. The efficacy of the model is also improved, since the percentage of retrieved documents remains lower when the number of documents increases.

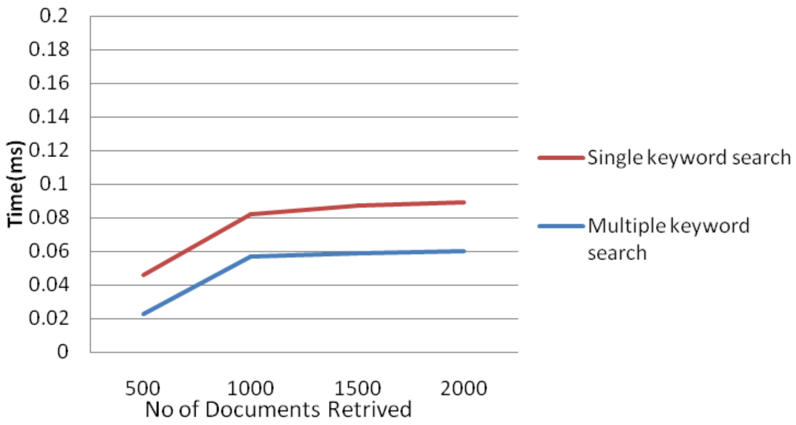

Figure 3 Multiple keyword search vs Single keyword search

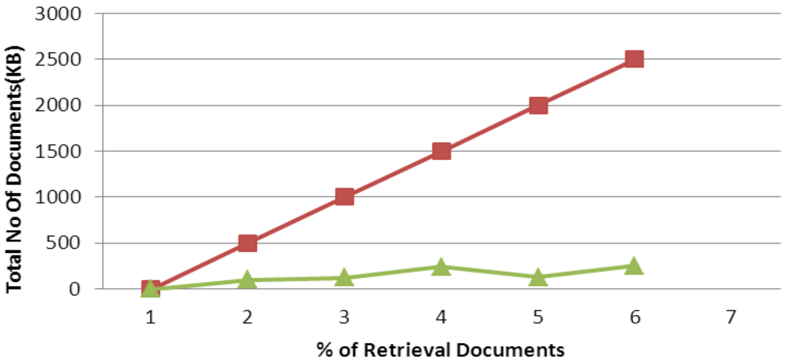

Figure 4 Storage overhead on the client side

6. CONCLUSION

The proposed framework allows the client to store documents in the cloud and at the same time enables the users to retrieve documents securely from the cloud. The search scheme proposed enables the server to produce queried documents and the corresponding documents are verified by the user with the keywords specified. The inverted index is constructed for documents with keywords and is encrypted along with documents using Hyper Elliptic Curve Cryptography. The Merkle tree is utilized for authentication of the index and the documents of the data owner as well as for the response from the server by the hash value at the top root. As data is outsourced, storage overhead on the client side is reduced and the user side also is less. This is because the necessary documents with the keywords are retrieved or else the entire document collection should be retrieved. The search scheme is efficient and correct under well-known assumptions, such as the hardness of discrete logarithmic problems and the security of the Merkle trees.

7. REFERENCES

Attrapadung, N., Libert, B., 2010. Functional Encryption for Inner Product: Achieving Constant-size Ciphertexts with Adaptive Security or Support for Negation. *In*: Proceedings of PKC, pp. 384–402

Boneh, D., Crescenzo, G.D., Ostrovsky, R., Persiano, G., 2004. Public Key Encryption with Keyword Search. *In*: Proceedings of EUROCRYPT, pp. 506–522

Cao, N., Wang, C., Li, M., Ren, K., Lou, W., 2011. Privacy-preserving Multi-keyword Ranked Search over Encrypted Cloud Data. *In*: Proceedings of IEEE INFOCOM, pp. 829–837

Cao, Q., Fujita S., 2014. Cost-effective Replication Schemes for Query Load Balancing in DHT-based Peer-to-peer File Searches. *Journal of Information Processing Systems*, Volume 10, pp. 628–645

Chai Q., Gong, G., 2012. Verifiable Symmetric Searchable Encryption for Semi-Honest-but-Curious Cloud Servers, *In*: IEEE International Conference on Communications (ICC), 2012, pp. 917–922

Chang, Y.C., Mitzenmacher, M., 2005. Privacy Preserving Keyword Searches on Remote Encrypted Data. *In*: Proceedings of ACNS, pp. 391–421

Cheng, R., Yan, J., Guan, C., Zhang, F., Ren, K., 2015. Verifiable Searchable Symmetric Encryption from Indistinguishability Obfuscation. *In*: Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, ASIA CCS '15, New York, NY, USA, ACM, pp. 621–626

Curtmola, R., Garay, J.A., Kamara, S., Ostrovsky, R., 2006. Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions. *In*: Proceedings of ACM CCS, pp. 79–88

Goh, E.J., 2003. Secure Indexes (2003), *Cryptology ePrint Archive*. Available online at http://eprint.iacr.org/2003/216

Hwang, Y., Lee, P., 2007. Public Key Encryption with Conjunctive Keyword Search and its Extension to a Multi-user system, *In*: Pairing, 2007

Kissel, Z. A., Wang, J., 2013. Verifiable Phrase Search over Encrypted Data Secure Against a Semi-honest-but-curious Adversary, *In*: IEEE 33rd International Conference on Distributed Computing Systems Workshops (ICDCSW), pp. 126–131

Lee, S. H., Lee I.Y., 2013. A Secure Index Management Scheme for Providing Data Sharing in Cloud Storage. *Journal of Information Processing Systems*, Volume 9, pp. 287–300

Liesdonk, P., Sedghi, S., Doumen, J., Hartel, P., Jonker, W., 2010. Computationally Efficient Searchable Symmetric Encryption. *Secure Data Management*, Volume 6358, pp. 87–100

Merkle, R. C., 1988. A Digital Signature Based on a Conventional Encryption Function. *In*: Advances in Cryptology–CRYPTO'87, Springer, pp. 369–378

Pang, H., Shen, J., Krishnan, R., 2010. Privacy-preserving Similarity-based Text Retrieval. *ACM Transactions on Internet Technology*, Volume 10(1), pp. 4:1–4:39

Salam, M.I., Yau, W.C., Chin, J.J., Heng, S.H., Ling, H.C., Phan, R.C.W., Poh, G.S., Tan, S.Y., Yap, W.S., 2015. Implementation of Searchable Symmetric Encryption for Privacy-preserving Keyword Search on Cloud Storage. *Human-Centric Computing and Information Sciences*, Volume 5(19), pp. 1–16

Shen E., Shi, E., Waters, B., 2009. Predicate Pivacy in Encryption Systems. *In*: Proceedings of TCC, pp. 457–473

Shi, E., Bethencourt, J., Chan, H., Song, D., Perrig A., 2007. Multi-dimensional Range Query over Encrypted Data. *In*: Proceedings of S & P, pp. 350–364

Song, D., Wagner, D., Perrig, A., 2000. Practical Techniques for Searches on Encrypted Data. *In*: Proceedings of S & P, pp. 44–55

Subashini, S., Kavitha, V., 2011. A survey on Security Issues in Service Delivery Models of Cloud Computing. *Journal of Network and Computer Applications*, Volume 34, pp. 1–11

Swaminathan, A., Mao, Y., Su, G.M., Gou, H., Varna, A.L., He, S., Wu, M., Oard, D.W., 2007. Confidentiality-preserving Rank-ordered Search. *In*: Proceedings of the 2007 ACM Workshop on Storage Security and Survivability, pp. 7–12

Wang, C., Cao, N., Ren, K., Lou, W., 2012. Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data. *IEEE Transactions on Parallel and Distributed Systems*, Volume 23(8), pp. 1467–1479

Xu, L., Weng, C.Y., Yuan, L.P., Wu, M.-E., Sun, H.M., Tso, R., 2015. A Shareable Keyword Search over Encrypted Data in Cloud Computing. *The Journal of Supercomputing*, pp. 1–23

Yang, J.J., Li, J.Q., Niu, Y., 2015. A Hybrid Solution for Privacy Preserving Medical Data Sharing in the Cloud Environment. *Future Generation Computer Systems*, pp. 74–86

Zerr, S., Olmedilla, D., Nejdl, W., Siberski, W., 2009. Zerber+R – Top-k Retrieval from a Confidential Index. *In*: Proceedings of EDBT, pp. 439–449

Zheng, Q., Xu, S., Ateniese, G., 2014. VABKS: Verifiable Attribute-based Keyword Search over Outsourced Encrypted Data. *In*: INFOCOM, 2014 Proceedings IEEE, pp. 522–530