Online Resource 2

1 Search Structures in Searchable Encryption

In Searchable Encryption (SE), a ciphertext comprises an encrypted payload along with an Index (search structure) of encrypted keywords. From literature, we determine that most of the existing verification enabled Symmetric Searchable Encryption (SSE) schemes [1–10] utilize Inverted Index (II) search structure in construction of searchable ciphertexts. On the other hand, there exist a few SSE schemes [11–16] with an alternate i.e. Simple Index (SI) search structure in ciphertext. In particular, II is a list of keyword values prepared by pre-processing the existing documents. With each keyword value in II, a list of document identifiers containing that keyword value is associated. In II-based SSE schemes, a single inverted index along with the collection of encrypted documents is uploaded onto server [17]. Subsequently, the search (for a single keyword) is performed across a single index. The search result includes a list of document identifiers associated with the keyword matching with the searched keyword. On the other hand, SI is a list of keyword fields where position of each field is pre-specified. Moreover, with each field of SI, a keyword value has been assigned. In SI-based SSE scheme, a separate simple index is associated with each encrypted document. The search is performed across index of each document separately and result includes the associated document for successful search. Note that a precise view of SI vs II is illustrated with example in Section 2. Formally, utilizing a single common index, II search structure seems storage efficient as compared to SI search structure. However, in practice SI has several benefits over II as discussed in Table 1.

Table 1: Simple Index vs. Inverted Index search structure

Simple Index (SI)	**Inverted Index (II)**
• SI is a separate index I_i associated with each data item $D_i \in D$ where $D = \{D_1, ..., D_m\}$.	• II is a single index I that is common amongst entire data collection $D = \{D_1, ..., D_m\}$.
• **SI supports dynamic data collection**. Since each D_i has its own I_i, the insertion of a new data item D_{m+1} does not affect the stored data collection D.	• **II supports static data collection** i.e. D must be available apriori to construct an index I. The insertion of D_{m+1} needs an index update operation [18–20].
• **SI supports variable values for n keyword fields** in index I_i=$\{w_{i1} = v_{i1}, ..., w_{in} = v_{in}\}$ where $v_{ij} \in V_{ij}$ $(1 \leq j \leq n)$ and V_{ij} is the set of possible values for keyword field w_{ij}.	• **II supports static values for keywords** in index i.e. I=$\{v_1, ..., v_V\}$ where V is a total number of keyword-values to be supported. As compared to SI, V=$\sum_{j=1}^{n} \lvert V_j \rvert >> n$.
• The **size of an index** I_i is constant i.e. $\lvert I_i \rvert$=$O(n)$ for $1 \leq i \leq m$. It is **comparatively smaller then the size of II**.	• For V different values, the size of an index is $\lvert I \rvert$=$O(V)$ that is **much higher than the size of SI**.
• An SI-based SSE scheme includes **one-phase search** algorithm where a search result contains data item D_i if I_i satisfies the search criteria.	• A majority of II-based SSE schemes include **two-phase search** algorithm where in the first phase, a search result includes list $L = \{ID_1, ..., ID_{N_s}\}$ $(1 \leq N_s \leq m)$ for N_s data items satisfying search criteria where ID_i is an identifier for data item D_i $(1 \leq i \leq N_s)$. In the second phase, user fetches actual data items D_{ID_i} from the server.
• Using SI, an **efficient conjunctive-keyword search** is possible with constant sized search token.	• With II, a **conjunctive-keyword search** involves either leakage of keyword-data relationship to server or exponential i.e. $O(2^V)$ storage overhead at server [12, 14] and hence **impractical**.

2 Application Scenario

We illustrate an application scenario where SI-based SSE scheme would be more effective than II-based SSE scheme and show the importance of result verification in case of malicious storage server.

Let us consider a firm of Certified Public Accountants (CPA) that collects financial information from the registered customers and prepares legal financial documents. To share data amongst employees (CPAs), the firm hires cloud storage

services. In addition, the firm utilizes an SSE scheme to securely search data stored on the cloud server. Figure 1 represents a scenario of CPA searching data.

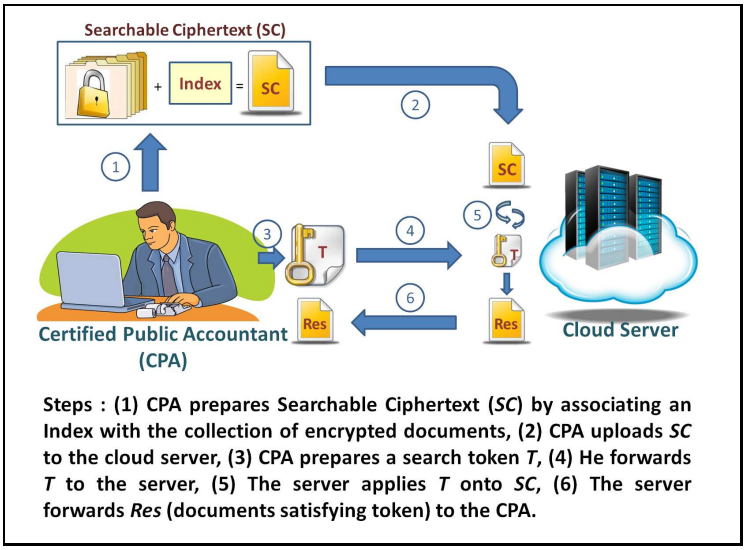

Figure 1: Scenario of a CPA

To precisely define the scenario, we demonstrate an SSE with Index of $n = 6$ keyword fields, i.e. I_n = (*CID, Rel, Day, Month, Year, ITRF*) where each keyword field $w_i \in I$ has different values as follows:

- $CID \in \{C001, C002...\}$ - *the ID assigned to customer by the firm*,
- *Rel* $\in$ {*ITR(Income Tax Return), AUDIT(Auditing of account), ST(Sales Tax), GST(Goods and Service Tax), PLO(Personal Loan), HLO(Housing Loan)...*} - *the relevancy of document*,
- $Day \in \{01, .., 31\}$, *Month* $\in$ {*Jan,...,Dec*}, $Year \in \{2016, 2017, ...\}$ - *Date on which document is prepared*,
- *ITRF* $\in$ {*Pending, Completed*} - *Whether ITR filing is pending or completed.*

Note that we consider the set of values for each keyword as V_{ij} (defined in Table 1).

To employ II search structure in construction of Searchable Ciphertext (SC), the firm requires a predefined collection of encrypted documents. Additionally, SC includes an II of size $O(\sum_{j=1}^{n} |V_j|)$ with prefixed set of keyword-values (Figure 2(a)). Moreover, once an SC is uploaded onto server, subsequent insertion of a new document into the uploaded data collection or insertion of a new keyword value into II, requires the computationally expensive index update operation linear to $O(V)$ [18–20]. On the other hand, for the underlined application, registration of a new customer and future data collection from the registered customers are the most common activities. In addition, with respect to the contents of documents, distinct values for keywords may be associated with each document. Therefore, II-based SSE scheme is impractical for the application under consideration.

On the other hand, with SI search structure, the firm needs to associate a comparatively small, fixed length index I (of size $O(n) \leq\leq O(\sum_{j=1}^{n} |V_j|)$) with variable values of keywords with each document as shown in Figure 2(b). In addition, since each new document has its own associated index, insertion of a document to the stored data collection does not incur any additional cost. With this fact, we infer that SI-based SSE scheme is indeed beneficial for the underlined application scenario.

Considering the advantages of SI, let us assume that the firm utilizes an SI-based SSE scheme and a CPA attempts the following queries:

1. *List documents for customer 'C*001*' related to 'ITR' for year '*2017*'.*
   *Q=('CID=C*001*' AND 'Rel=ITR' AND 'Year=*2017*')*

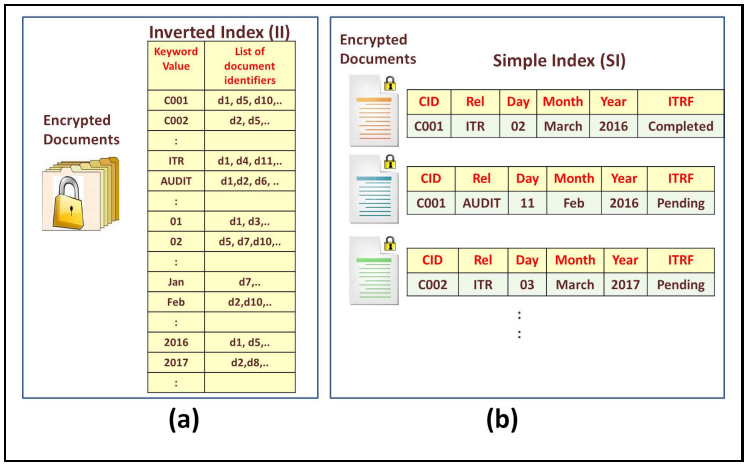

Figure 2: Searchable Ciphertexts (SC)

2. *List documents for year '2017' where 'ITR' is pending.*
   *Q=('Year=*2017*' AND 'Rel=ITR' AND 'ITRF=Pending')*

3. *List documents for year '2017' where filing of 'ITRF' is completed.*
   *Q=('Year=*2017*' AND 'Rel=ITR' AND 'ITRF=Completed')*

For each query, server performs conjunctive search operation across Index and returns either encrypted payload data (for successful search) or NULL (for unsuccessful search) as a search result. In such a setup, if server is malicious, it may deliberately alter the search result either by tampering the resultant data or by forwarding a NULL value for successful search. For the listed queries, any alteration of result, may lead CPA to the erroneous information about the status of income tax return file. Therefore, verification of the returned result is indeed essential in case of a malicious cloud server. Additionally, it should be noted that conjunctive keyword search is a desirable operation for the above queries.

References

[1] Jianfeng Wang, Xiaofeng Chen, Hua Ma, Qiang Tang, Jin Li, and Hui Zhu. A verifiable fuzzy keyword search scheme over encrypted data. *Journal of Internet Services and Information Security (JISIS)*, 2:49–58, 2012.

[2] HweeHwa Pang and Kyriakos Mouratidis. Authenticating the query results of text search engines. *Proceedings of the VLDB Endowment*, 1(1):126–137, 2008.

[3] HweeHwa Pang and K-L Tan. Authenticating query results in edge computing. In *Data Engineering, 2004. Proceedings. 20th International Conference on*, pages 560–571. IEEE, 2004.

[4] Feifei Li, Marios Hadjieleftheriou, George Kollios, and Leonid Reyzin. Dynamic authenticated index structures for outsourced databases. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 121–132. ACM, 2006.

[5] Yanbin Lu. Privacy preserving logarithmic-time search on encrypted data in cloud. In *NDSS*, 2012.

[6] Qi Chai and Guang Gong. Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers. In *Communications (ICC), 2012 IEEE International Conference on*, pages 917–922. IEEE, 2012.

[7] Kaoru Kurosawa and Yasuhiro Ohtaki. Uc-secure searchable symmetric encryption. In *International Conference on Financial Cryptography and Data Security*, pages 285–298. Springer, 2012.

[8] Zachary A Kissel and Jie Wang. Verifiable phrase search over encrypted data secure against a semi-honest-but-curious adversary. In *Distributed Computing Systems Workshops (ICDCSW), 2013 IEEE 33rd International Conference on*, pages 126–131. IEEE, 2013.

[9] Wenhai Sun, Bing Wang, Ning Cao, Ming Li, Wenjing Lou, Y Thomas Hou, and Hui Li. Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking. *IEEE Transactions on Parallel and Distributed Systems*, 25(11):3025–3035, 2014.

[10] Rong Cheng, Jingbo Yan, Chaowen Guan, Fangguo Zhang, and Kui Ren. Verifiable searchable symmetric encryption from indistinguishability obfuscation. In *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, pages 621–626. ACM, 2015.

[11] Eu-Jin Goh et al. Secure indexes. *IACR Cryptology ePrint Archive*, 2003:216, 2003.

[12] Philippe Golle, Jessica Staddon, and Brent Waters. Secure conjunctive keyword search over encrypted data. In *Applied Cryptography and Network Security*, pages 31–45. Springer, 2004.

[13] Yan-Cheng Chang and Michael Mitzenmacher. Privacy preserving keyword searches on remote encrypted data. In *International Conference on Applied Cryptography and Network Security*, pages 442–455. Springer, 2005.

[14] Lucas Ballard, Seny Kamara, and Fabian Monrose. Achieving efficient conjunctive keyword searches over encrypted data. In *Information and Communications Security*, pages 414–426. Springer, 2005.

[15] Jin Wook Byun, Dong Hoon Lee, and Jongin Lim. Efficient conjunctive keyword search on encrypted data storage system. In *European Public Key Infrastructure Workshop*, pages 184–196. Springer, 2006.

[16] Peishun Wang, Huaxiong Wang, and Josef Pieprzyk. Keyword field-free conjunctive keyword searches on encrypted data and extension for dynamic groups. In *Cryptology and Network Security*, pages 178–195. Springer, 2008.

[17] Dawn Xiaodong Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on*, pages 44–55. IEEE, 2000.

[18] Qiang Tang. Search in encrypted data: Theoretical models and practical applications. *Theory and Practice of Cryptography Solutions for Secure Information Systems*, 84, 2013.

[19] Hongwei Li, Dongxiao Liu, Yuanshun Dai, and Tom H Luan. Engineering searchable encryption of mobile cloud networks: When qoe meets qop. *IEEE Wireless Communications*, 22(4):74–80, 2015.

[20] Yunling Wang, Jianfeng Wang, and Xiaofeng Chen. Secure searchable encryption: a survey. *Journal of communications and information networks*, 1(4):52–65, 2016.