ISSN: 2366-1313

New Lightweight Symmetric Searchable Encryption Scheme for String Identification

Y HARIKA, Assistant professor
Dr. SIKHAKOLLI GOPI KRISHNA, professor
CSE Department, Sri Mittapalli College of Engineering, Guntur, Andhra Pradesh-522233

Abstract—In this paper, we provide an efficient and easy-to-implement symmetric searchable encryption scheme (SSE) for string search, which takes one round of communication, O(n) times of computations over n documents. Unlike previous schemes, we use hash-chaining instead of chain of encryption operations for index generation, which makes it suitable for lightweight applications. Unlike the previous SSE schemes for string search, with our scheme, the server learns nothing about the frequency and the relative positions of the words being searched except what it can learn from the history. We are the first to propose probabilistic trapdoors in SSE for string search. We provide concrete proof of the nonadaptive security of our scheme against honest-but-curious servers based on the definitions of [12]. We also introduce a new notion of search pattern privacy, which gives a measure of security against the leakage from the trapdoor. We have shown that our scheme is secure under the search pattern indistinguishability definition. We show why the SSE scheme for string search cannot attain adaptive indistinguishability criteria as mentioned in [12]. We also propose modifications to our scheme so that the scheme can be used against active adversaries at the cost of more rounds of communications and memory space. We validate our scheme against two different commercial datasets (see [1], [2]).

Index Terms—Cloud storage, Symmetric key, Searchable encryption, hash-chain, lightweight cryptography.

I. INTRODUCTION

The cloud is designed to hold a large number of encrypted documents. With the advent of cloud computing, a growing number of clients and leading organizations have started adapting to private storage outsourcing. This allows resource-constrained clients to privately store large amounts of encrypted data in the cloud at a low cost. However, this prevents one from searching. This gives rise to a newly emerging field of research, called searchable encryption (SE). SE can be classified into symmetric searchable encryptions (SSE) and asymmetric searchable encryptions (ASE). In this paper, we study the SSE for string search. In the SSE, the client encrypts the data and stores it in the cloud. It may be noted that the client can organize the data in an arbitrary manner and can maintain additional data structures to achieve desired data efficiently. In this process, the initial client-side computation is thus as large as the data, but subsequent computations to access data are less for both client and the cloud server. Since huge volumes of documents are stored in a cloud server, searching against a keyword may result in a large number of documents, most of which are not intended, causing unnecessary network traffic. This motivates the idea of searching against a string, which allows the search to be more specific. Searching for a string is a multi-keyword search where the ordering of keywords is preserved. So in addition to the presence of all these keywords in a document, their ordering and adjacency are to be taken care of while searching. The index table needs to be prepared in such a way that the adjacency information of the words can be preserved. Although few works are available in the literature involving string search, but most of them lack formal security proof against the revised definitions of [12] and also expose lots of information to the server following the search (see Table I of Section II). In the SSE scheme, the server is expected to learn nothing about the search queries and data collections. SSE achieves this by using symmetric

ISSN: 2366-1313

cryptographic primitives instead of heavy computations of public key encryption at the cost of small leakage of information [12]. Here we take an example which will be extended throughout the paper to illustrate our algorithms and data structures.

II. RELATED WORKS

For the last ten years, searchable encryption has been the focus for many leading research groups and several results were proposed, authors defined computational and statistical relaxations of the existing notion of perfect consistency and provided a new scheme that was statistically consistent. They also proposed a transformation of an anonymous identity based encryption scheme (IBE) to a secure public key encryption with keyword search scheme (PEKS) that guarantees consistency. In [4] authors presented as-strong-as-possible definitions of privacy and some constructions for public-key base encryption schemes where the encryption algorithm is deterministic. In the same work, new methods were proposed for database encryption that permit fast (i.e. sub-linear, and in fact logarithmic, time) search while provably providing privacy that is as strong as possible subject to this fast search constraint. The work in [4] also generalizes their methods to obtain a notion of efficiently searchable encryption schemes which permit more flexible privacy to search-time trade-offs via a technique called bucketization. In [5], authors studied the problem of searching on data that is encrypted using a public key system which they referred as PEKS and provided several constructions. In [6], authors show how to create a public-key encryption scheme that allows PIR (private information retrieval) searching over encrypted documents. Their solution was the first to reveal no partial information regarding the users search (including the access pattern) in the public-key setting and with small communication complexity. In [7], authors defined and solved the problem of privacy-preserving multi-keyword ranked search over encrypted data in cloud computing (MRSE). They established a set of strict privacy requirements for such a secure cloud data utilization system. Among various multi-keyword semantics, they choose the efficient similarity measure of ”coordinate matching,” i.e., as many matches as possible, to capture the relevance of data documents to the search query. They also used ”inner product similarity” to quantitatively evaluate such similarity measure. They provide two MRSE schemes to achieve various stringent privacy requirements in two different threat models. In [24], authors proposed an efficient searchable encryption scheme for auction (SESA) in emerging smart grid marketing, which is based on a public key encryption with keyword search technique to enable the energy sellers to inquire suitable bids while preserving the privacy of the energy buyers. In [8] authors provided a systematic study of various attack models against SSE based schemes. Dynamic SSE was first considered by Song et al. [19], but no solution with sublinear search time existed before the work of Kamara et al. [13]. Recently, two new dynamic SSE schemes have been proposed. The first one, by Cash et al. [9], which is an extension of [10]. They showed that SSE is feasible on very large databases. In [9], authors designed and implemented dynamic symmetric searchable encryption schemes that efficiently and privately search serverheld encrypted databases with tens of billions of record-keyword pairs. Their basic theoretical construction was for single-keyword searches and which offers asymptotically optimal server index size, fully parallel searching, and minimal leakage. In [10], authors presented another efficient SSE scheme which supports complex queries involving multiple keywords. Similar scheme may be found. In [11], authors studied the trade-off between locality and server storage size of SSE schemes. In [12], authors introduced the idea of SSE with improved security definitions. They introduced the two most important security definitions, namely non-adaptive indistinguishability and adaptive indistinguishability. They also proposed SSE schemes for keyword search which they proved to be secure under these new security definitions. In [16] authors studied the security provided by various encrypted databases and presented a series of attacks that recover the plaintext from encrypted database columns using only the encrypted column and publicly-available auxiliary information.

ISSN: 2366-1313

authors studied efficient sub linear search techniques for arbitrary Boolean queries. They considered scalable DBMS with provable security for all parties, including protection of the data from both server (who stores encrypted data) and client (who searches it), as well as protection of the query, and access control for the query.

III. Proposed methodology

The proposed a SSE scheme for string search, which is similar to the non-adaptive SSE scheme of [12] for keyword search with some additional data structures and techniques (list, lookup tables, pseudo random functions and hash-chains for word sequencing) being used to keep track of position informations. However, with this approach, server learns the word frequency and relative positions of the underlying document. The BuildIndex algorithm (see Algorithm 2) is based on a new approach of inverted index generation in modulo prime field. As opposed to the unmasked hash-chains used in [15] for all words in a document, we use masked hash-chain and cell-padding which stops leakage of informations related to the relative positions of sentences and the frequency of words. As opposed to the idea of chain of encryptions in [12], we introduce the idea of masking for the security of index which is faster. In earlier schemes, to search for a word in n documents, n decryption operations were needed. In our scheme, all we need is unmasking which is a subtraction operation in Zp for all entries of the corresponding column.

A. Key Generation

Algorithm 1 Keygen

Input security parameter λ.
Output k_m, k', k_s and p.
$k_m, k', k_s \leftarrow Gen(1^\lambda)$;
$p \leftarrow PPNG(1^\lambda)$;

Algorithm 3 Trapdoor

Input $w = (w_{c_1}, w_{c_2}, \ldots, w_{c_l})$, k_m, k' k_s, p, $MAC_k(.)$.
Output $t = (t_1, \ldots, t_l)$.
$j \leftarrow 1$;
while $j \leq l$ **do**
$e = Enc_{k_m}(w_{c_j})$;
$msk = MAC_{k'}(w_{c_j})$;
$ci = MAC_{k_m}(w_{c_j})$;
$t_{j_1} = MAC_{k_s}(e \oplus msk \oplus ci)$;
$t_{j_2} = e \otimes ci$;
$t_{j_3} = e \otimes msk$;
$t_j = (t_{j_1}, t_{j_2}, t_{j_3})$;
$j \leftarrow j + 1$;
end while

ISSN: 2366-1313

B. Index Generation

Algorithm 2 BuildIndex

Input k_m, k', k_s, p, $D = (D_1, \ldots, D_n)$.
Output $SI = (I, I_r, I_c)$.

Form a collection $W = \{w_1, \ldots, w_d\}$ of all distinct words occurring in D;
$j \leftarrow 1$;
while $j \leq d$ **do**
  $ci_j = MAC_{k_m}(w_j)$;
  $I_r[j] = ci_j^{-1}$;
  $j \leftarrow j + 1$;
end while
$i \leftarrow 1$;
while $i \leq n$ **do**
  $I_c[i] \leftarrow Enc_{k_m}(id(D_i))$;
  For each sentence $s = (w_{s_1}, \ldots, w_{s_l})$ in D_i, chose $r \in \mathbb{Z}_p$ randomly and form $(r_1, \ldots, r_l)$ such that $r_1 = r$ and for $2 \leq j \leq l$, $r_j = MAC_{k_s}(r_{j-1})$. Associate r_j with the word w_{s_j}.
  $j \leftarrow 1$;
  while $j \leq d$ **do**
    set $I[i][j]$ as all integers in $\mathbb{Z}_p$ that are associated with the word w_j and add the mask $m_j = MAC_{k'}(w_j)$ with all of them in modulo p, i.e., in $\mathbb{Z}_n$;
    if $(|I[i][j]| < f)$ **then**
      Inject $(f - |I[i][j]|)$ number of random elements from $\mathbb{Z}_p^*$ in $I[i][j]$;
    end if
    $j \leftarrow j + 1$;
  end while
  $i \leftarrow i + 1$;
end while

ISSN: 2366-1313

Algorithm 4 Search

Input $t = (t_1, \ldots, t_l)$, SI, k_s, $MAC_k(.)$.
Output $encrypted\_file\_pointers$, a list of encrypted document pointers;
the list $column$ and $column\_msk$ are set empty;
$i \leftarrow 1$;
while $i \leq l$ **do**
  $j \leftarrow 1$;
  while $j \leq d$ **do**
    $e = (t_{i_2} \otimes I_r[j])$;
    $m = t_{i_3} \otimes e^{-1}$;
    if $(MAC_{k_s}\left(e \oplus m \oplus I_r[j]^{-1}\right) == t_{i_1})$ **then**
      set mask of j th column as $msk = m$, add j to $column$ and msk to $column\_msk$;
    end if
    $j \leftarrow j + 1$;
  end while
  $i \leftarrow i + 1$;
end while
the list $encrypted\_file\_pointers$ is set empty;
$i \leftarrow 1$;
while $i \leq n$ **do**
  if (there exists l integers $p_1, \ldots, p_l$ such that $(p_j \oplus column\_msk[j]) \in I[i][column[j]]$, $1 \leq j \leq l$ and $MAC_{k_s}(p_j) == p_{j+1}$ for $1 \leq j \leq l-1$) **then**
    add $I_c[i]$ to $encrypted\_file\_pointers$.
  end if
  $i \leftarrow i + 1$;
end while

An honest-but-curious server follows the protocol and takes no actions beyond those of an honest server, and attempts to learn about the plaintext of documents or terms that were queried. The idea of non-adaptive security for SSE scheme for an honest-butcurious server was first introduced in [12]. In order to explain the non-adaptive security, we first provide the definition of history and trace.

VII. CONCLUSION

With the increasing number of documents stored in the cloud, searching for the desired document can be a difficult and resource-intensive task. One solution may be to use symmetric searchable encryption (SSE) which allows one party to outsource the storage of its data to another party (a cloud) privately while enabling to search selectively over it. In this paper we revisited the security

ISSN: 2366-1313

definitions of [12] and proposed a new lightweight SSE scheme Πs,s for string search. We have shown that our scheme is secure under the non-adaptive indistinguishability definition [12]. For active adversary, we propose a modification of the scheme Πs,s at the additional cost of memory at client's end and two rounds of communications for one modification of document collection. Towards this direction, future research can be performed to design efficient SSE scheme ideally with one round of communication. With our scheme, server does not learn the information related to word frequency and word positions except what it can learn from the history. We, for the first time, introduce new security notion in SSE, named, search pattern indistinguishability. It may be observed that with non adaptive indistinguishability security, although the keywords are guaranteed to be secure from the possible leakage from index, however it does not guarantee the security from the possible leakage from trapdoor. Towards this, we for the first time introduce probabilistic trapdoor and prove that our scheme is secure under such criterion. We have implemented our scheme for the first time to search over phone symbols and validated it using the TIMIT dataset. We have also implemented our scheme over DNA data of [2] and successfully achieve pattern matching functionality over encrypted domain. While dealing with string search, designing a SSE scheme satisfying adaptive-indistinguishability-security definition of [12] seems intuitively impossible. This is because to generate an index in advance which is consistent with future search, one unavoidable assumption needed is the presence of all possible strings in each document. This can be done by considering all permutations of keywords for every document which makes the index size exponential in n for ndocument collection. According to the definition of [12], index size is linear in n which is essential from efficiency point of view. From the angle of this intuition, future research can be carried out to give a formal proof in support of non-existence of adaptively secure SSE scheme for string search. In this paper we have considered honestbut-curious adversaries and active adversaries. Also, designing SSE scheme for string search with adaptive-indistinguishability-security against some newly defined adversary can be a future research direction.

REFERENCES

[1] http://www.fon.hum.uva.nl/david/ma ssp/2007/timit/train/dr5/fsdc0/. 2007.

[2] https://github.com/iskana/pbwt-sec/tree/master/sample dat. 2015.

[3] Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions. volume 21, pages 350–391. Springer, 2008.

[4] Mihir Bellare, Alexandra Boldyreva, and Adam ONeill. Deterministic and Efficiently Searchable Encryption. In Annual International Cryptology Conference, pages 535–552. Springer, 2007.

[5] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public Key Encryption With Keyword Search. In International Conference on the Theory and Applications of Cryptographic Techniques, pages 506–522. Springer, 2004.

[6] Dan Boneh, Eyal Kushilevitz, Rafail Ostrovsky, and William E Skeith III. Public Key Encryption That Allows PIR Queries. In Annual International Cryptology Conference, pages 50–67. Springer, 2007.

[7] Ning Cao, Cong Wang, Ming Li, Kui Ren, and Wenjing Lou. PrivacyPreserving Multi-Keyword Ranked Search Over Encrypted Cloud Data. volume 25, pages 222–233. IEEE, 2014.

ISSN: 2366-1313

[8] David Cash, Paul Grubbs, Jason Perry, and Thomas Ristenpart. LeakageAbuse Attacks Against Searchable Encryption. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pages 668–679. ACM, 2015.

[9] David Cash, Joseph Jaeger, Stanislaw Jarecki, Charanjit S Jutla, Hugo Krawczyk, Marcel-Catalin Rosu, and Michael Steiner. Dynamic Searchable Encryption in Very-Large Databases: Data Structures and Implementation. volume 2014, page 853. Citeseer, 2014.

[10] David Cash, Stanislaw Jarecki, Charanjit Jutla, Hugo Krawczyk, MarcelCatˇ alin Ros¸u, and Michael Steiner. Highly-Scalable Searchable Sym- ˇ metric Encryption With Support for Boolean Queries. In Advances in Cryptology–CRYPTO 2013, pages 353–373. Springer, 2013.

[11] David Cash and Stefano Tessaro. The Locality of Searchable Symmetric Encryption. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 351–368. Springer, 2014.

[12] Reza Curtmola, Juan Garay, Seny Kamara, and Rafail Ostrovsky. Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions. volume 19, pages 895–934. IOS Press, 2011.

[13] Seny Kamara, Charalampos Papamanthou, and Tom Roeder. Dynamic Searchable Symmetric Encryption. In Proceedings of the 2012 ACM conference on Computer and communications security, pages 965–976. ACM, 2012.

[14] Jonathan Katz and Yehuda Lindell. Introduction to Modern Cryptography. CRC press, 2014.

[15] Mingchu Li, Wei Jia, Cheng Guo, Weifeng Sun, and Xing Tan. LPSSE: Lightweight Phrase Search With Symmetric Searchable Encryption in Cloud Storage. In Information Technology-New Generations (ITNG), 2015 12th International Conference on, pages 174–178. IEEE, 2015.

[16] Muhammad Naveed, Seny Kamara, and Charles V Wright. Inference Attacks on Property-Preserving Encrypted Databases. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pages 644–655. ACM, 2015.

[17] Vasilis Pappas, Fernando Krell, Binh Vo, Vladimir Kolesnikov, Tal Malkin, Seung Geol Choi, Wesley George, Angelos Keromytis, and Steve Bellovin. Blind Seer: A Scalable Private DBMS. In 2014 IEEE Symposium on Security and Privacy, pages 359–374. IEEE, 2014.

[18] Hoi Ting Poon and Ali Miri. Fast phrase search for encrypted cloud storage. IEEE Transactions on Cloud Computing, doi: 10.1109/TCC.2017.2709316, 2017.