International Journal of Research
Available at https://pen2print.org/index.php/ijr/
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 06 Issue 1
January 2019

A Novel Symmetric Searchable Encryption Scheme for String Identification in Cloud Computing

Mrs.Rizvana Shaik & Mr.P.Srinivasulu
1PG Scholar, Dept of CSE, MalineniLakshmaiah Engineering College, Singarayakonda, Prakasam(Dt), AP, India.
2Associate Professor, Dept of CSE, MalineniLakshmaiah Engineering College, Singarayakonda, Prakasam(Dt), AP, India.

Abstract_ *In this paper, we first provide an efficient and easy-to-implement symmetric searchable encryption scheme (SSE) for string search, Unlike previous schemes, we use hash chaining instead of chain of encryption operations for index generation, which makes it suitable for lightweight applications. Here it's first to propose probabilistic trapdoors in SSE for string search. We provide concrete proof of non-adaptive security of our scheme against honest-but-curious server based on the definitions . Here also introduce a new notion of search pattern privacy, which gives a measure of security against the leakage from trapdoor. i have shown that our scheme is secure under search pattern in distinguishability definition. i show why SSE scheme for string search cannot attain adaptive in distinguishability criteria as mentioned and also propose modifications of our scheme so that the scheme can be used against active adversaries at the cost of more rounds of communications and memory space.*

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

I. INTRODUCTION

The cloud is intended to hold a substantial number of encoded reports. With the approach of distributed computing, developing number of customers and driving associations have begun adjusting to the private stockpiling re-appropriating. This permits asset obliged customers to secretly store a lot of scrambled information in cloud effortlessly. Be that as it may, this keeps one from seeking. This offers ascend to a recently rising field of research, called accessible encryption (SE). SE can be ordered into symmetric accessible encryptions (SSE) and lopsided accessible encryptions (ASE). In this paper, we think about the SSE for string look. In the SSE, the customer encodes the information and stores it on the cloud. It might be noticed that customer can arrange the information in a self-assertive way and can keep up extra information structures to accomplish wanted information proficiently. In this procedure, the underlying customer side calculation is in this way as expansive as the information, however resulting calculations to

get to information is less for both customer and the cloud server.

Since enormous volumes of archives are put away in a cloud server, seeking against a catchphrase may result into extensive number of documents, the vast majority of which are not expected, causing superfluous system traffic. This rouses looking against a string, which enables the inquiry to be progressively explicit. Looking for string is a multi catchphrase seek where the requesting of watchwords is protected. So notwithstanding the nearness of every one of these catchphrases in a report, their requesting and contiguousness are to be taken consideration off while searchingthe record table should be set up so that the nearness data of the words can be protected.

Albeit few works are accessible in the writing including string look (for example [1], [8], [2], [3], [5]), however the vast majority of them need formal security evidence against the overhauled meanings of [1] and furthermore uncover heaps of information's to the server following the pursuit In the SSE conspire, the server is required to get the hang of nothing about the inquiry inquiries and information accumulations. SSE accomplishes this by utilizing symmetric cryptographic natives rather than overwhelming calculations of open key encryption at the expense of little spillage of data [1]. Here we take a model which will be reached out all through the paper to outline our calculations and information structures

III. Literature Survey

Mihir Bellare, Alexandra Boldyreva, and Adam ONeill. Deterministic and Efficiently Searchable Encryption. In Annual International Cryptol-ogy Conference, pages 535–552. Springer,

The classical notions of privacy for public-key encryption schemes, mainly indistinguishability or semantic security under chosen-plaintext or chosen-ciphertext attack [4,3,8,10], can only be met when the encryption algorithm is randomized. This paper treats the case where the encryption algorithm is deterministic. We begin by discussing the motivating application. Fast search. Remote data storage in the form of outsourced databases is of increasing interest [2]. Data will be stored in encrypted form. (The database service provider is not trusted.) We are interested in a public key setting, where anyone can add to the database encrypted data which a distinguished "receiver" can retrieve and decrypt. The encryption scheme must permit search (by the receiver) for data retrieval. Public-key encryption with keyword search (PEKS) [11,1,8] is a solution that provably provides strong privacy but search takes time linear in the size of the database. Given that databases can be terabytes in size, this is prohibitive. The practical community indicates that they want record containing a given field value can be a retrieved in time logarithmic in the size of the database. (For example, via appropriate tree-based

data structures.) Deterministic encryption allows just this. The encrypted fields can be stored in the data structure, and one can find a target ciphertext in time logarithmic in the size of the database. The question is what security one can expect. To answer this, we need a definition of privacy for deterministic encryption. A definition. One possibility is to just ask for one-wayness, but we would like to protect partial information about the plaintext to the maximum extent possible. To gauge what this could be, we note two inherent limitations of deterministic encryption. First, no privacy is possible if the plaintext is known to come from a small space. Indeed, knowing that c is the encryption under public key pk of a plaintext x from a set X, the adversary can compute the encryption cx of x under pk for all x ∈ X, and return as the decryption of c the x satisfying cx = c. We address this by only requiring privacy when the plaintext is drawn from a space of large min-entropy. Second, and more subtle, is that the ciphertext itself is partial information about the plaintext. We address this by only requiring non-leakage of partial information when the plaintext and partial information do not depend on the public key. This is reasonable because in real life public keys are hidden in our software and data does not depend on them. While certainly weaker than the classical notions met by randomized schemes, our resulting notion of privacy for deterministic encryption, which we call PRIV, is still quite strong. The next question is how to achieve this new notion. Constructions. Our first construction is generic and natural: Deterministically encrypt plaintext x by applying the encryption algorithm of a randomized scheme but using as coins a hash of (the public key and) x. We show that this "Encrypt-with-Hash" deterministic encryption scheme is PRIV secure in the random oracle (RO) model of [11] assuming the starting randomized scheme is IND-CPA secure. Our second construction is an extension of RSA-OAEP [10,1]. The padding transform is deterministic but uses three Feistel rounds rather than the two of OAEP. RSA-DOAEP is proven PRIV secure in the RO model assuming RSA is one-way. This construction has the attractive feature of being length-preserving. (The length of the ciphertext equals the length of the plaintext.) This is important when bandwidth is expensive — senders in the database setting could be power-constrained devices— and for securing legacy code.

IV. OUR SCHEME

In this section we present our SSE scheme $_{ss}$ for string search, which is composed of four algorithms KeyGen, BuildIndex, Trapdoor and Search. First we formally define the scheme. In the subsequent subsections, we will discuss these algorithms in detail with illustra-tions.

Scheme 1 ($_{ss}$). The scheme $_{ss}$ is a collection of four polynomial time algorithms (KeyGen, BuildIndex, Trapdoor, Search) such that:

1. KeyGen(1) : KeyGen is a probabilistic key generation algo-rithm that is run by the

client to setup the scheme (see Algorithm 1). It takes a security parameter , and returns a secret master key k_m and a mask-key k^0 which are to be kept privately at client's end and a session key k_s which is to be shared between client and the server. Client also shares a -bit prime p with the server. The length of k_m, k^0 and k_s are polynomially bounded

2. BuildIndex(k_m; k^0; k_s; p) : BuildIndex is a probabilistic algo-rithm run by the client to generate SI = (I; I_r; I_c). It takes k_m, k^0, k_s, p and returns SI. Since BuildIndex is randomized, we write this as SI $\text{BuildIndex}_{(km;k0;ks;p)}(s)$.

3. Trapdoor(k_m; k_s; p; s) :Trapdoor is a probabilistic algorithm run by the client to generate a trapdoor for a given string of words s = (w_1; w_2; : : : ; w_l). It takes k_m, k^0, k_s, p and s as input and outputs t = (t_1; t_2; : : : ; t_l), where t_i is the trapdoor corresponding to the word w_i. Since trapdoor is randomized, we write this as t $\text{T rapdoor}_{(km;k0;ks;p)}(s)$

4. Search(SI; t) : Search is run by the server in order to search for the documents in D that contain the string s. It takes k_s, SI and trapdoor t of the string s as inputs, and returns D(s), the set of identifiers of documents containing the string s. Since this algo-rithm is deterministic, we write it as $D(s) = \text{Search}_{ks}$ (SI; t)

4.Conclusion

With the increasing number of documents stored in cloud, search-ing 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 definitions of [11] and proposed a new lightweight SSE scheme $_{s;s}$ for string search. We have shown that our scheme is secure under the non-adaptive in distinguishability definition [11]. For active adversary, we propose modification of the scheme $_{s;s}$ 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 in distinguishability. It may be observed that with

non adaptive in distinguishability security, although the keywords are guar-anteed 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 [1] and successfully achieve pattern matching functionality over encrypted domain

REFERENCES

[1] https://github.com/iskana/pbwt sec/tree/master/sample dat.

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

[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 Proper-ties, 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 Cryptol-ogy Conference, pages 535–552. Springer, 2007.

[5] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public Key Encryption With Keyword Search. In Interna-tional 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 Interna-tional Cryptology Conferencē, pages 50–67. Springer, 2007.

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

[8] David Cash, Paul Grubbs, Jason Perry, and Thomas Ristenpart. Leakage-Abuse 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 Search-able Encryption in Very-Large Databases: Data Structures and Imple-mentation. volume 2014, page 853. Citeseer, 2014.

[10] David Cash, Stanislaw Jarecki, Charanjit Jutla, Hugo Krawczyk, Marcel-Cat˘alin˘ Ros¸u, and Michael Steiner. Highly-Scalable Searchable Sym-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.

Author's Profile

Mr.P.Srinivasulu Has Received His B.Tech In CSE In And M.Tech PG In Computer Science From JNTU, Hyderabad 2008. He Is Dedicated To teaching Field From The Last 12 Years. He Has Guided 15 P.G Students And 28 U.G Students. His is a Member in MISTE. His Research Included Data Mining At Present He Is Working As Assoc Professor And HOD In Malineni Lakshmaiah Engineering College, Singarayakonda, Prakasam(Dt), AP, India.. He Is Highly Passionate And Enthusiastic About Her Teaching And Believes That Inspiring Students To Give Of His Best In Order To Discover What He Already Knows Is Better Than Simply Teaching.

Mrs.RIZVANA SHAIK received B.Tech in Computer Science and Information Technology from St. Ann's College of Engineering & Technology,Chirala, affiliated to Jawaharlal Nehru Technological University, Andhra Pradesh in 2005.

She has nine years of working experience in IT field as a software consultant and currently pursuing M.TECH in computer science and engineering from Malineni Laxmaiah college of engineering and technology affiliated to Jawaharlal Nehru Technological University, Kakinad in 2017-2019.