ISSN: 2321-7782 (Online)
Volume 2, Issue 1, January 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Paper
Available online at: www.ijarcsms.com

A Secure Database by using SAD Algorithm

Vrushali S. Sakharkar[1]
Prmit & R, Badnera
India

Dr. G. R. Bamnote[2]
Head Of Department
Prmit & R, Badnera
India

Abstract: ***The ability to securely share sensitive information between untrusting parties is a prerequisite for many real-world applications. A general methodology for this is using searchable encryption techniques, which allows encrypted data to be searched by users without leaking information about the data itself and user's queries. By using a new primitive re-routable encryption along with Bloom filters and deterministic encryption, SADS lets multiple parties efficiently execute exact-match queries over distributed encrypted databases in a controlled manner. A secure anonymous database search (SADS) system that provides exact keyword match capability. Re- routable encryption has received considerable attention in applications where private and sensitive data about users can be stored in un-trusted database (DB) servers. It allows users to store encrypted data at DB servers and yet retain the ability to search those databases without revealing anything else about the encrypted data. Traditional SADS lacks flexibility and semantic awareness of the corpora over which it operates. The exact-match constraint that traditional SADS imposes doesn't allow for search capabilities such as grammatical case and number. To overcome this limitation, the notion of context-specific semantically aware feature extraction is applied to encrypted search scenarios. SADS is extended in two ways: Its search capabilities beyond exact keyword match and provide a modular framework for adapting the system to meet varying security and efficiency needs.***

Keywords: Database Privacy, Re-routable Encryption, Semantic, SADS, Search on Encrypted Data.

I. INTRODUCTION

Existing systems for encrypted search provide privacy guarantees, but at a provably high cost in efficiency. As such, these systems scale poorly and aren't suitable for real-world applications with very large flexibility and semantic awareness of the corpora over which it operates. SADS system uses third parties and relaxed definitions of security to circumvent these inherent efficiency costs. Data encryption guarantees data confidentiality, it also rules out many routine manipulations over the data necessary in the plaintext domain. One fundamental requirement is to be able to perform search operations that can sort out relevant information from huge amounts of data. For enterprise end users, database search is an everyday operation that underlies their corporate business intelligence. Over the past decade, SAD technique have become a significant research area , which are tailored cryptographic solutions addressing privacy assured search over encrypted data under different system requirements and security models. It is possible for functional usability and efficiency to be simultaneously achieved in order to build privacy-assured searchable data services. It is started by identifying an important set of desirable properties including both privacy goals and search usability. A general methodology for constructing privacy-assured symmetric-key encryption primitives. For each of the proposed usable search functionalities, a survey is made on recent research search schemes based on several building blocks, including recently developed efficient advances and give insights on the advantages and limitations of each approach. It focuses on how to enable privacy-assured search for data services. The system architecture captures a wide range of searchable data storage applications. Simple keyword matching alone isn't always enough to satisfy real-world queries over complex, structured documents. To overcome the constraints that SADS imposes, both the document corpus and all incoming queries are preprocessed to achieve greater semantic awareness and improved search flexibility while operating under

the exact-match constraint. The general framework presented goes beyond textual corpora; the feature extraction process can be applied to any document format.

II. System Requirement

The system is based on the SADS protocol, which offers a model to achieve balance among the conflicting requirements of security, anonymity, and practical efficiency. In particular, it obtains query efficiency that scales sub-linearly with the number of search entries in a document and number of possible search terms.

- Privacy Assurance:

  More specifically, the system should meet the following privacy requirements:

- Data and index confidentiality:

  Without the secret key K, no one, including the index server should be able to learn sensitive information from the owner's private data. Similarly,one should not be able to deduce sensitive information underlying the data index, because the index is often closely related to the data itself.

- Query confidentiality:

  User's most important concern is to hide the search criteria on which they are evaluating the data (e.g., their query keywords). These should not be derivable from the search trapdoor and data/index sent to the index server, even when the server possesses some additional background information such as keyword distribution. A higher-level requirement is query unlinkability, that is, the Index server shall not learn whether two queries have the same criteria. This intrinsically requires the trapdoor to be non-deterministic.

- Efficiency:

  A privacy-assured data search scheme should have low computation, communication, and storage overheads. For such a scheme to be deployed in a large-scale index storage system with economic practicality, the search process should be completed within both constant communication round and computation time (independent of the database size).

III. Methodology and Building Blocks

The following building-blocks protocols are used to construct SADS system.

- Re-routable encryption: An encryption system that would allow encryption transformation under given corresponding keys without leaking information is needed. In re-routable encryption, one party is responsible for routing messages between senders and receivers. Although the routing party is trusted to match senders and receivers, it's not trusted to learn the routed messages. For this, the routing party is allowed to let to forward only partial information extracted from the transformed ciphertext.

- PH-DSAEP+: Whereas the re-routable encryption system provides a framework to realize the QR functions, an additional property that would facilitate efficient search in the IS is needed. Because the standard cryptography definitions of security require an encryption system to be probabilistic, which makes sublinear search complexity impossible, deterministic encryption is needed. PH-SAEP+ is a combination of the Pohlig-Hellman function and the SAEP+ (Simple-OAEP) padding construction introduced in "Simplified OAEP for the RSA and Rabin Functions,"and PH-DSAEP+ with the removal of the nondeterministic component.

- Bloom filters: The private-key deterministic encryption system PH-DSAEP+ provides search capability over cipher texts, achieving sub-linear complexity. To utilize this capability, searchable tags from deterministic encryptions are constructed. For this, Bloom filters are used which extend the idea of hashing using multiple hash functions, improving collision

probabilities. This facilitates efficient search, which requires constant time per Bloom filter, which guarantees that there won't be false negatives, and allows a tunable rate of false positives. A Bloom filter is computed for each document in the database. Each Bloom filter contains all keywords from a single document, and on those keywords PHDSAEP+ is used to generate hash function indexes.

IV. SADs ALGORITHM

Key generation

Both the server and the IS choose an encryption key. The client generates a key for query submission and a key for retrieving results. To authorize the client to search the server, the QR and the client run a key exchange protocol to let the QR obtain aratio key between the server's encryption key and the client's query submission key. Also, the IS, client, and QR run a key-exchange protocol so that the QR obtains a ratio key between the IS's encryption key and the client's return result key.

Preprocessing:

The server generates a Bloom filter for each of its documents by encrypting all words in the document using PH-DSAEP+ under its own key. The server sends the resulting Bloom filters to the IS.

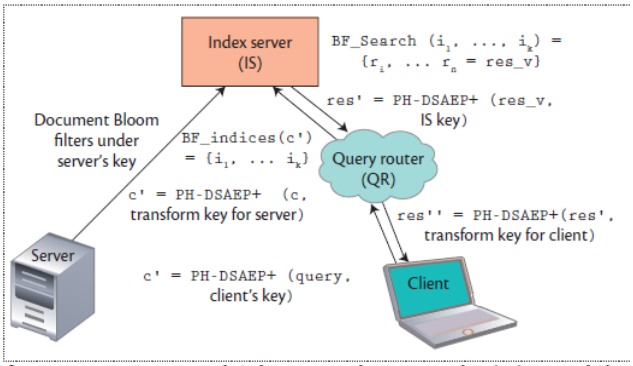

Figure 1. A secure anonymous database search query submission and the results. The server stores an encrypted document index on the IS. The client can then send encrypted and anonymized queries to the IS through the QR.

Query submission:

The client encrypts its query with PH-DSAEP+ with its key and sends it to the QR. The QR reencrypts the ciphertext with its transformation key for the client, extracts Bloom filter indexes from the new encryptions, and sends them to the IS. Search. The IS uses the obtained indexes to execute the Bloom filter search to get the result.

Query return:

The query result is returned with a different instantiation of the reroutable encryption protocol. The IS encrypts the result with PH-SAEP+ and sends it to the QR. The QR re-encrypts the ciphertext with the return result transformation key and sends it to the client. Achieving Secure Ranked Search over encrypted data an especially important functionality in plaintext IR is to support ranking mechanisms over search results according to user-specified relevance criteria.

TRADE-OFFS AND IMPACTS OF DESIGN CHOICES

In designing privacy-assured search schemes, it is generally perceived that efficiency vs. privacy is an intrinsic trade-off; acceptable to leak, higher efficiency may be gained without decreasing privacy, which depends on all the factors including the IR method, index structure, and encryption primitive. The implementation incorporates three well-known and currently used decision functions and produces the same results, barring false positive results added by the underlying Bloom filter–based implementation of the search. This is a tunable parameter and can be reduced to negligible levels at the cost of storage space at the system designer's discretion.

V. FUTURE CHALLENGES

System designers must consider the desired security and privacy requirements at a semantic level and carefully consider the optimal tolerance for semantic false positive and false negative errors, which is often a non-trivial compromise. There are many interesting research issues worth further investigation. The works mentioned above have a common characteristic: they relax the privacy guarantees to achieve higher efficiency performance. While there are formal privacy definitions for searchable encryption that reveal the access pattern , for as-strong-as possible schemes, how to formally analyze the privacy level given various known background information remains an interesting and important open problem.

VI. CONCLUSION

The problem and challenges of enabling privacy-assured searchable data storage services is identified. Recent research advances in this field are surveyed, which suggest that achieving functionally rich, usable and efficient search on encrypted data is possible without sacrificing privacy guarantee too much. The matching behavior to comply with high-level security and privacy requirements at a semantic level is correctly tuned for private secure search system.

References

1. M. Armbrust et al., "Above the Clouds: A Berkeley View of Cloud Computing,Feb 2009.
2. A. Boldyreva et al., "Order-Preserving Symmetric Encryption," Proc. Eurocrypt 09, LNCS, vol. 5479, Springer , 2009.
3. N. Cao et al., "Privacy-Preserving Multi-Keyword Ranked Search Over Encrypted Cloud Data," IEEE INFOCOM, 2011, pp. 829–37
4. A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," LNCS 3494, Springer, 2005, pp. 457–473.
5. M. Raykova et al., "Secure Anonymous Database Search," Proc. ACM Cloud Computer Security Workshop 09, ACM, 2009, pp. 115–126.
6. S. Goldwasser and S. Micali, "Probabilistic Encryption," J. Computer and System Sciences, vol. 28, no. 2, 1984 , pp. 270–299.

