A Machine-Learning-Based Approach to Suppress Volume Leakage
Project Proposal for Class CS 507

Jiaming Yuan

Sunday 21st January, 2024

Abstract

Searchable Encryption (SE) enables secure keyword queries on cloud-stored documents while preserving user privacy. However, adversaries exploiting leakages and auxiliary knowledge pose a threat known as Leakage Abuse Attack (LAA). A prevalent type of leakage, known as volume leakage, encompasses both query volume (indicating the number of documents related to a keyword) and document volume (indicating the number of keywords related to a document).

Under the consideration of the complexity of effectively suppressing these two forms of volume leakage, this project aims to develop a machine-learning-based approach. This innovative strategy seeks to mitigate volume leakage in searchable encryption, offering an advanced and efficient solution with k-indistinguishability.

Notably, this project is an individual undertaking for the CS 507 course, providing a focused exploration into the realm of privacy preservation in cloud-based searches.

1 Background

Searchable Encryption (SE) has been extensively investigated for enabling users to query keywords on a set of documents stored in a cloud platform while preserving privacy. This is achieved by outsourcing encrypted documents and associated keywords to the cloud, allowing users to search for keywords by sending tokens. However, adversaries, although unable to directly access information from encrypted documents, encrypted keywords, and tokens, can infer sensitive data through various leakages and auxiliary knowledge. This form of attack is known as Leakage Abuse Attack (LAA). To thwart such attacks, it is crucial to eliminate utilized leakages from a searchable encryption scheme, a process known as leakage suppression.

The primary challenge of leakage suppression is to mitigate leakages with limited storage overhead. For instance, padding dummy messages and documents to create only one leakage pattern for each type of leakage can result in significant storage overhead. This presents a trade-off between storage overhead and the number of leakage patterns. In our prior research, we introduced k-indistinguishability to address this, ensuring there are at least k records for each pattern of each type of leakage. For instance, if $k = 10$ for document volume leakage, there are at least 10 documents with a volume of 6.

Another challenge involves suppressing multiple types of leakages simultaneously. For instance, to hide volume leakage, both document volume and query volume leakages must be concealed simultaneously. Additionally, LAAs may leverage multiple types of leakage, including volume leakage and co-occurrence leakage.

To tackle these challenges, we proposed a machine-learning-based approach to hide leakages with k-indistinguishability. In this project, our focus is on hiding volume leakage, as it is the most frequently targeted by LAAs.

2 Problem Statement

We represent the relationships between a set of documents D_n and a set of associated keywords K_m through a matrix $M_{m\times n}$. In this matrix, rows correspond to keywords, columns correspond to documents, and $M_{i,j}$ indicates whether the keyword kw_i is associated with the document d_j. For a given keyword kw_i, its response within the matrix M is calculated as $R(kw_i, M) = \{d_j | M_{i,j} = 1, \forall j \in [1, n]\}$.

Volume Leakage. The query volume leakages are represented by all sums of rows of M, which is calculated as $L_{qv}(M) = \{\sum_{j=1}^{n} M_{i,j}\}_{i\in[1,m]}$. Similarly, the document volume leakages are represented by all sums of columns and calculated as $L_{dv} = \{\sum_{i=1}^{m} M_{i,j}\}_{j\in[1,n]}$.

Obfuscated Matrix. Given k, it produces an obfuscated matrix ${M'}_{m'\times n'}$ with a set of obfuscated documents $D'_{n'}$ and a set of obfuscated keywords $K'_{m'}$, where $D'_{n'} \supseteq D_n$ and $K'_{m'} \supseteq K_m$. For any $i \in [1, m]$ and $i' \in [1, m']$, if $kw_i = kw_{i'}$, then $R(kw_{i'}, M') \supseteq R(kw_i, M)$. This ensures that the response set for any keyword $kw_{i'}$ in the obfuscated matrix M' is a superset of the response set for the corresponding keyword kw_i in the original matrix M. In addition, for any value $v \in L_{qv}(M')$, it occurs at least k times in $L_{qv}(M')$. The same condition holds for $L_{dv}(M')$.

Problem. Given that the number of documents constitutes the majority of the storage overhead, the project problem is how to identify an obfuscated matrix ${M'}_{m'\times n'}$ with minimized n'.

2.1 Project Objectives.

The objectives of this project encompass designing a machine-learning-based approach to conceal volume leakage while relatively optimizing the number of obfuscated documents. Additionally, the project aims to conduct experiments using this approach on a sample set of documents and keywords, comparing the results with our prior approaches in terms of storage overhead and computation overhead.

Outcomes. This project offers practical exposure to the implementation, training, and fitting of machine learning models

3 Methodology

Building on our previous research, the strategy to conceal leakages involves the use of dummy documents, resulting in $K'_{m'} = K_m$. Additionally, we employ the modulo-based partition method to mask document volume and the keyword clustering method to conceal query volume—both techniques introduced in our prior research work.

The focus of the machine-learning-based approach is to seamlessly integrate these two methods while minimizing storage overhead overlap.

4 Deliverables

1. A GitHub repository to maintain the code `https://github.com/yjm9110/ml_vlr`.
2. A report on the detailed approach and experimental results.
3. Slides for the project presentation.

5 Timeline

Weeks	Task
Week 1	Explore potential machine learning models for leakage suppression.
Week 2	Identify and finalize the machine learning model for the project.
Week 3	Dedicate time to thoroughly understand the chosen machine learning model.
Week 4	Design the machine-learning-based method for volume leakage suppression.
Week 5	Refine the proposed method based on initial design considerations.
Week 6	Document the detailed construction of the machine-learning-based approach.
Week 7	Implement the designed method.
Week 8	Conduct experiments to evaluate the storage and computation overheads of the approach.
Week 9	Compile the final project report.
Week 10	Prepare presentation slides.

Table 1: Project Timeline