The statistical nature of leakage in SSE schemes and its role in passive attacks

MARC DAMIE, Inria, France and University of Twente, The Netherlands
JEAN-BENOIST LEGER, Université de technologie de Compiègne, CNRS, Heudiasyc, France and Université Paris-Saclay, AgroParisTech, INRAE, UMR MIA Paris-Saclay, France
FLORIAN HAHN, University of Twente, The Netherlands
ANDREAS PETER, Carl von Ossietzky Universität Oldenburg, Germany and University of Twente, The Netherlands

Encrypted search schemes have been proposed to address growing privacy concerns. However, several leakage-abuse attacks have highlighted the shortcomings of these schemes. The literature remains vague about the consequences of these attacks for real-world applications: are these attacks dangerous in practice? Is it safe to use these schemes? Do we even need countermeasures?

This paper introduces a novel mathematical model for attackers' knowledge using statistical estimators. Our model reveals that any attacker's knowledge is inherently noisy, which limits attack effectiveness. This inherent noise can be considered a security guarantee, a natural attack mitigation. Capitalizing on this insight, we develop a risk assessment protocol to guide real-world deployments. Our findings demonstrate that limiting the index size is an efficient leverage to bound attack accuracy. Finally, we employ similar statistical methods to enhance attack analysis methodology. Hence, our work offers a fresh perspective on SSE attacks and provides practitioners and researchers with novel methodological tools.

CCS Concepts: • **Security and privacy → Cryptanalysis and other attacks**; *Management and querying of encrypted data.*

Additional Key Words and Phrases: Searchable Encryption, Attacks, Risk assessment, Statistics

1 INTRODUCTION

With the increasing popularity of cloud data storage services, there is a growing concern about the privacy and confidentiality issues induced by such practices. Song et al. [49] proposed the first construction to search on encrypted data. Curtmola et al. [11] later used this result to build a Searchable Symmetric Encryption (SSE) scheme. This scheme enables an efficient keyword search across encrypted documents.

Such SSE schemes build an encrypted index that can be queried to obtain the (encrypted) documents containing a given keyword. These schemes leak information exploitable by an attacker to recover the plaintext query. Our work studies the single-keyword search SSE schemes leaking access and search patterns. Our focus will be on the type of attack that received the most attention: passive attacks assuming attacker-known documents [6, 12, 14, 21, 26, 36, 42, 43, 46, 57]. These attacks build a co-occurrence matrix from the leakage and compare it to a keyword co-occurrence matrix computed on an attacker-known document set.

Passive query-recovery attacks. There are two types of passive query-recovery attacks against SSE: similar-data attacks and known-data attacks. The known-data attacks assume that the attacker-known documents are indexed, while similar-data attacks only require an attacker's knowledge "similar but different" from the indexed data [1, 12]. Hence, similar-data attacks cover a more general setting. The first attack papers [6, 26, 46] proposed attacks theoretically usable as similar-data attacks but accurate only as known-data attacks. Several works [1, 36, 40, 41, 44] later proposed attacks that are only usable as known-data attacks. Damie et al. [12] and Oya and Kerschbaum [43] respectively presented the

Authors' addresses: Marc Damie, marc.damie@inria.fr, Inria, Villeneuve d'Ascq, France and University of Twente, Enschede, The Netherlands; Jean-Benoist Leger, Université de technologie de Compiègne, CNRS, Heudiasyc, Compiègne, France and Université Paris-Saclay, AgroParisTech, INRAE, UMR MIA Paris-Saclay, Palaiseau, France; Florian Hahn, University of Twente, Enschede, The Netherlands; Andreas Peter, Carl von Ossietzky Universität Oldenburg, Oldenburg, Germany and University of Twente, Enschede, The Netherlands.

refined score attack and the *IHOP attack*, which both achieved high recovery rates as a similar-data attack. Recently, Gui et al. [21] introduced another similar-data attack robust to several attack countermeasures.

Other types of attacks exist, but they are out of scope: active attacks [58, 59], attacks exploiting the query frequency [37, 42], attacks against other encrypted search schemes [20, 30, 35].

Security assessment of SSE schemes. All the attack papers successively improved state-of-the-art. The attack analysis methodology is the same in all attack papers: the authors simulate multiple attacks with varying numbers of attacker-known documents. Then, the accuracy is represented as a function of the number of attacker-known documents. However, the literature gives no tool to evaluate their efficiency in real-world scenarios.

On the one hand, the literature does not explain whether the simulation parameters used in attack papers are realistic. For example, all papers generate the (attacker-known and indexed) document sets by uniformly splitting a research dataset (e.g., Enron [31]). We do not know whether this default choice leads to over- or under-estimated attack results compared to a real-world attacker.

On the other hand, simulations (even using a research dataset from real-world activity) do not tell whether an SSE deployment over another not-previously-analyzed dataset is at risk. For example, Damie et al. [12] showed that attack simulations on different datasets can provide different attack accuracies. Hence, the existence of a risk on specific research datasets does not imply that the attacks are dangerous for all possible datasets/use cases.

Therefore, estimating the *real-world risk* induced by the existing attacks represents a significant gap in SSE literature. The attacker must know a "similar enough" dataset to achieve a successful attack. We must understand the difficulty of finding such datasets to properly assess the real-world potential of SSE attacks.

Related works. Kamara et al. [27] developed a Python framework to standardize the attack simulation commonly used in attack papers. Their standardization is essential to have a consistent attack comparison. However, it does not provide more information about the practicality of the attacks than the existing papers. Kornaropoulos et al. [34] proposed a "privacy quantification", which quantifies the size of the attacker reconstruction space for a given leakage. Their approach has two main shortcomings. First, the reconstruction space size does not provide much information about the attack accuracy. A small reconstruction space mechanically induces more successful attacks. However, an attack can also be successful with a large reconstruction space: within this space, some solutions might be more likely than others. Second, their work only considered "exact knowledge" (e.g., the attacker knows perfectly the keyword frequency in the indexed document set), which is an unrealistic assumption. Indeed, recent attacker papers [1, 6, 12, 21, 43, 46] all considered an "approximate knowledge" (e.g., an attacker-known document set). The extension to "approximate knowledge" is listed as future work in [34]. Concurrently to our work, Kamara and Moataz [28] used (different) statistical tools to analyze the leakage in encrypted search. While our work studies commonly implemented schemes (i.e., with access and search pattern leakage), they focus on schemes with less leakage. Moreover, our work builds upon a novel model representing the attacker's knowledge as statistical estimators. Dittert et al. [15] is another concurrent work. They explore experimentally the link between document set similarity and attack accuracy, already highlighted by [12]. We share the same initial intuition as Dittert et al. [15]: having similar datasets is non-trivial. However, they neither provide a mathematical model to analyze this phenomenon nor provide actionable results.

Our contributions. Our paper presents a novel statistical approach to understand SSE leakage. Our analysis reveals that the attacker's knowledge, including the scheme leakage itself, is inherently noisy, which limits the attack's effectiveness. This statistical perspective provides valuable insights into the conditions necessary for a successful attack. Furthermore,

Table 1. Summary of recurring notations

Notation	Meaning	Size
D_{atk}	Attacker-known documents	n_{atk}
D_{ind}	Indexed documents	n_{ind}
$\mathcal{W}$	Keyword universe	m
C^X	Co-occurrence matrix built from D_X	$m \times m$
p_{ij}^X	$\mathbb{P}(\text{keywords } (i,j) \text{ appear in } d \xleftarrow{R} D_X)$	N.A.

our mathematical model bridges the gap between simulation-based attack results and the real-world impact of these attacks. Concretely, our paper contains three main contributions:

(1) A novel *mathematical model for the attacker's knowledge* based on statistical estimators (Section 3). This model highlights that the SSE security is conditioned by the hardness of estimating precisely some statistics on the indexed data.
(2) A statistical method to *assess the risk* of deploying an SSE scheme in an organization, assuming the knowledge of a sample dataset (Section 4). This method can estimate a maximum index size under which the attack accuracy is negligible. This contribution aims to help practitioners deploy SSE schemes with a controlled risk.
(3) A framework based on a similarity metric to provide *a consistent and more interpretable attack analysis and comparison* (Section 5). We provide several conclusions about the parameters influencing attack accuracy.

We develop our analysis for similar-data attacks against static SSE schemes with access and search pattern leakage. Section 6 discusses the extensions of our analysis to other schemes (e.g., schemes with less leakage) and attacks.

2 PRELIMINARIES

Table 1 summarizes the most used notations of the paper. Our notations take inspiration from [12] as our setting (i.e., similar-data attacks) is closely related to their setting.

2.1 Searchable symmetric encryption (SSE)

From a high-level point of view, SSE schemes are based on the design described in [11]. The client owns a document set D_{ind}. For each $d \in D_{\text{ind}}$, we denote $\text{id}(d)$ its identifier. We assume that $\text{id}(\cdot)$ leaks no information about d. The client generates an inverted index and encrypts it using her secret key K. Then, this encrypted index is uploaded to the server. A query token is an output of the query function (denoted as $\text{Query}_K(w)$) taking as input the keyword w and the secret key K. This index associates query tokens with the identifiers of the documents containing the underlying keyword. To query the keyword w, the client computes $\text{Query}_K(w)$ and sends this token to the server. The server sends back the matching encrypted documents.

This work focuses on efficient SSE constructions [3, 5, 7, 8, 11, 17, 50, 51] that leak the search and access patterns. Blackstone et al. [1] proposed the avoidance of access pattern leakage, but its instantiation is inefficient. Hence, our paper only studies SSE schemes leaking search and access patterns. This leakage profile has been the target of many passive attacks over the last decade [6, 12, 14, 21, 26, 36, 42, 43, 46, 57].

2.2 Threat model

A passive attacker can observe and link each query to its response. Thus, the attacker can create (Query, DocIDs) pairs for each query observed. The access pattern leakage corresponds to the list of the matching (encrypted) documents leaked by each query. The search pattern leakage lets an attacker identify if a query has been issued twice. Hence, each keyword w has a unique query token $\text{Query}_K(w)$.

We study the case of an *honest-but-curious* server storing the encrypted index and following the protocol while trying to recover the keywords being queried. Such an attacker can record and analyze the protocol transcript. Nearly all attack papers used this setting; only [58, 59] proposed active attacks. Finally, the attacker can use leaked data as long as she is not actively involved in the leak (e.g., a leaked dataset published by hackers).

Almost all papers presenting passive attacks studied SSE schemes supporting single-keyword search in a static setting (i.e., no update operation). On the one hand, only Dijkslag et al. [14] studied attacks on (static) schemes supporting conjunctive keyword search. On the other hand, only Xu et al. [57] presented a passive attack in a dynamic setting. Therefore, our paper focuses on static schemes with single-keyword search because we inherit the assumptions of state-of-the-art attacks. Section 6 discusses the extensions of our work to different settings.

2.3 Attacker's knowledge

Let $\text{Coocc}(D, w_a, w_b)$ be the function returning the number of documents from D in which both w_a and w_b appear (i.e., the number of co-occurrences).

Keyword universe. The keyword universe $\mathcal{W} = \{w_1, \ldots, w_m\}$ is the set of keywords that the user can query. Like most papers, we assume that the attacker knows the keyword universe.

Similar document set. The attacker knows a document set $D_{\text{atk}} = \{d_1, \ldots, d_{n_{\text{atk}}}\}$. The documents are "*similar but different*" from the indexed documents. Similar documents can be, for example, documents that have been leaked (then removed from the index). From D_{atk}, the attacker obtains the $m \times m$ keyword co-occurrence matrix defined as follows: $C_{ij}^{\text{atk}} = \text{Coocc}(D_{\text{atk}}, w_i, w_j)$.

Observed queries. The attacker has observed l unique queries $\mathcal{Q} = \{q_1, \ldots, q_l\}$ and their respective results. We denote as $R(q) = \{id(d) | d \in D_{\text{ind}} \wedge q = \text{Query}_K(w) \wedge w \in d\}$ the list of document identifiers returned for the query q. The attacker can then compute the $l \times l$ query co-occurrence matrix $C_{ij}^{\text{query}} = \left|R(q_i) \cap R(q_j)\right|$.

Known queries. In [6, 12, 26], the attacker has some *known queries*. In other words, for k different queries, she knows the underlying keyword of the query.

Number of indexed documents. Finally, the attacker knows the number of documents indexed n_{ind}. An honest-but-curious server storing the index can infer this metadata.

2.4 Attacks

This paper demonstrates statistical methods on state-of-the-art attacks (i.e., the Score, Refined Score [12], and IHOP [43] attacks), but our insights hold for all passive query-recovery attacks on SSE. Any future attack could be analyzed analogously. Our results concern the attack analysis methodology but not directly the attacks themselves. The attack algorithm *takes as input* the attacker's knowledge presented in Subsection 2.3. The algorithm *outputs* a predicted keyword for each observed query. The attack accuracy is the proportion of correctly predicted keywords.

Refined score attack. This attack [12] uses the co-occurrence matrices and the known queries to extract a vector characterizing each keyword and each query. Each keyword vector is characterized by its co-occurrence with the keywords from the known queries. Each query vector is characterized by its co-occurrence with the queries from the known queries. In this vectorization, a keyword pair should be close to its corresponding query vector. The score attack identifies, for each query, the keyword minimizing the keyword-query distance (based on their respective vectors). The refined score attack improves this naive idea via an iterative process to improve the query recovery.

IHOP attack. The IHOP (Iteration Heuristic for quadratic Optimization Problems) attack [43] also uses the co-occurence matrices but does not require known queries contrary to [12]. This paper formulates the attack problem as an assignment problem (i.e., assigning queries to keywords). The objective function to minimize is a sum of cost functions (i.e., one per query). The problem is NP-Complete because it is a quadratic optimization problem. The main contribution of [43] is to solve this problem using an efficient iterative algorithm based on linear assignment. While the refined score attack uses a vector distance to quantify the cost of an assignment, IHOP relies on log-likelihood functions.

2.5 Experimental setup

We perform our experiments on Debian Bullseye with a 4-core processor and 8 GB of memory. We use three datasets: the commonly used Enron email dataset [31] (30,109 emails contained from *_sent_mail* folders), the Apache mailing dataset [1] (the 50,878 emails from the "java-user" mailing list from the Lucene project for 2002-2011), and the Blog Authorship dataset [2] [48] (681,288 blog posts written by 19,320 authors), never used in an existing attack paper. Due to hardware constraints, we only use 200K posts simultaneously (picked uniformly at random) from this last dataset.

Since we are studying similar-data attacks, the dataset is split into two disjoint subsets of varying sizes to create D_{atk} and D_{ind}. By default, we perform this split uniformly at random, but in Subsection 3.5, we present several results using a split based on the email timestamp.

We only extract keywords from the document content. The keyword list is then obtained after having stemmed the words using the Porter stemmer [45] and removed the stop words (e.g., "the" or "and"). In the Apache dataset, we remove the mailing list signature from each email. To generate the keyword universe $\mathcal{W}$, we choose the m most frequent keywords in the complete dataset. We use keyword universes of varying sizes from $m = 500$ to $m = 4K$.

We measure the attack accuracy as the proportion of correct keyword recovery among a set of *unique* observed queries. In other words, we do not consider duplicate queries.

Our Python codebase is publicly available here: https://github.com/MarcT0K/Statistical-Leakage-SSE-attacks. We refactored the source code of the attacks published by [12, 43][3].

3 THE ATTACKER'S KNOWLEDGE, A NOISY KNOWLEDGE

This section describes the statistical nature of leakage and highlights the direct consequences on attack success. Subsection 3.2 model the attacker's knowledge using statistical estimators, while Subsections 3.3 and 3.4 leverage this mathematical model to explore the influence of document set sizes (especially the index size) on knowledge quality and attack accuracy. Finally, Subsection 3.5 questions the attacker's knowledge generation commonly used in attack papers. In particular, we show that the commonly used algorithm simulates an unrealistic attack scenario, but highlight its interest for conservative risk assessment.

[1] http://mail-archives.apache.org/mod_mbox/lucene-java-user/
[2] Dataset archive: https://web.archive.org/web/20200121222642/http://u.cs.biu.ac.il/~koppel/blogs/blogs.zip
[3] Source code: https://github.com/MarcT0K/Refined-score-atk-SSE, https://github.com/simon-oya/ihop-code

3.1 Document set similarity

Damie et al. [12] introduced the notion of document set similarity to quantify the divergence between the attacker-known data and the indexed data.

Let C^{atk} (resp. C^{ind}) be an $m \times m$ co-occurrence matrix defined as follows: $C^{\text{atk}}_{ij} = \text{Coocc}(D_{\text{atk}}, w_i, w_j)$ (resp. $C^{\text{ind}}_{ij} = \text{Coocc}(D_{\text{ind}}, w_i, w_j)$) for all keywords $w_i, w_j \in \mathcal{W}$. The query co-occurrence matrix C^{query} is a restriction of C^{ind} with an unknown rotation (the goal of the attacks is to find the rotation): if $q_a = \text{Query}_K(w_i)$ and $q_b = \text{Query}_K(w_j)$ then $C^{\text{query}}_{ab} = C^{\text{ind}}_{ij}$. Damie et al. [12] define the $m \times m$ similarity matrix of D_{ind} and D_{atk} over the keyword universe $\mathcal{W}$ as follows:

$$\text{SimMat} = \frac{C^{\text{ind}}}{n_{\text{ind}}} - \frac{C^{\text{atk}}}{n_{\text{atk}}} \tag{1}$$

Definition 1. *The document sets D_{ind} and D_{atk} are ϵ-similar if* $||SimMat|| \leq \epsilon$

Definition 1 uses the Frobenius norm (i.e., the equivalent of Euclidean norm for matrices) as matrix norm. In the rest of the paper, we refer to ϵ as the smallest value that satisfies this inequality (i.e., $\epsilon = \|\text{SimMat}\|$).

Alternative metrics. Gui et al. [21] introduced two other similarity metrics: "absolute distance" and "Modified Probability Score". The absolute distance has the same formula as the ϵ-similarity but uses the infinity norm instead of the Frobenius norm. The Modified Probability Score relies on conditional probabilities.

Metric choice. There is no universally better metric; the choice of metric is subjective. Our paper uses the ϵ-similarity because its formula is convenient for mathematical analysis. In other contexts, the Modified Probability Score might be preferable: e.g., if the link between the metric and conditional probability is necessary for mathematical analysis.

Attacker similarity assumption. All passive attacks using attacker-known data make (at least implicitly) a similarity assumption concerning the attacker's knowledge. The attacker assumes that there exists a sufficiently small ϵ such that D_{ind} and D_{atk} are ϵ-similar. If we assume no similarity, the attacker could use random data.

Dissecting this assumption and its implications is necessary to assess the practicality of SSE attacks. More precisely, we want to answer the question: **how likely is it to obtain a "sufficiently similar" document set for a given use case?** This question represents a fundamental starting point for this paper.

3.2 Revisiting the notion of similar data

As in machine learning, we consider a dataset as a sample of a random distribution. The properties of the underlying probability distribution could bring more or less uncertainty depending on the use case. We will model this uncertainty and show how it contributes to SSE security.

Mathematical model. The first step toward understanding the similarity assumption is establishing a proper mathematical model for the co-occurrence matrices. Let $x \xleftarrow{R} \mathcal{X}$ denote the sampling of x from the random probability distribution $\mathcal{X}$.

Let D_{atk} and D_{ind} be two document sets composed of documents that are represented as binary vectors of length m. Each vector component i indicates whether the keyword w_i is contained in the document. Furthermore, let $\mathcal{X}_{\text{atk}}$ denote the random variable that describes the experiment of sampling a document (i.e., a binary vector of length m) from the same probability distribution as given by the documents in D_{atk}. In other words, $\mathcal{X}_{\text{atk}}$ is a vector of dependent

Bernoulli random variables. It implies that the co-occurrence matrix C^{atk} is composed of realizations of dependent Binomial variables, where $C^{\text{atk}}_{ij} \xleftarrow{R} \mathcal{C}^{n_{\text{atk}},p_{\text{atk}}}_{ij} = \mathcal{B}(n_{\text{atk}}, p^{\text{atk}}_{ij})$. The probability p^{atk}_{ij} corresponds to the probability that the keywords i and j both appear in a document from D_{atk}. Acknowledging the dependence between the Binomial variables is essential because it significantly complicates the mathematical analysis. To be convinced of the dependence, let us consider the probabilities p_{ij} and p_{ii} (i.e., the probability of keyword i appearing in a document). We have $p_{ij} \leq p_{ii}$ because the event "keyword i appears in a document" contains the event "keywords i and j appears in the same document".

Let $\mathcal{X}_{\text{ind}}$ be the random variable for document set D_{ind} defined with the same procedure as $\mathcal{X}_{\text{atk}}$ for document set D_{atk}. Similarly, we deduce that the co-occurrence matrix C^{ind} is composed of realizations of dependent Binomial variables, where $C^{\text{ind}}_{ij} \xleftarrow{R} \mathcal{C}^{n_{\text{ind}},p_{\text{ind}}}_{ij} = \mathcal{B}(n_{\text{ind}}, p^{\text{ind}}_{ij})$.

To sum up, $\mathcal{C}^{n_{\text{ind}},p_{\text{ind}}}$ and $\mathcal{C}^{n_{\text{atk}},p_{\text{atk}}}$ are two random matrix probability distributions from which C^{ind} and C^{atk} are respectively drawn. We assume the two distributions to be independent because, in similar-data attacks, the attacker document set is considered "different": implicitly, it means obtained from independent sources.

Statistical estimators. The previously defined random distribution ($\mathcal{C}^{n_{\text{ind}},p_{\text{ind}}}$ and $\mathcal{C}^{n_{\text{atk}},p_{\text{atk}}}$) are unknown to the attacker (or even the researcher). We are in a classic statistics problem: we do not have access to the exact probabilities, but we know a dataset drawn from an *unknown* probability distribution. We can use the dataset to compute an *estimation* of an *unknown* probability. The interest of statistical estimation is then to approximate these unknown probabilities.

Finding a similar document set is close to the problem of representative sampling for surveying: the results are not the probabilities themselves (e.g., political opinions) but an estimation of this probability. The larger the sample is, the more precise the estimation. If one compares two samples obtained from populations with different probabilities, their respective estimators may be close to each other, but it is unlikely. For example, a survey on professors might provide similar results to the same survey on students. However, one understands that it is unlikely, especially when the number of questions in the survey grows (i.e., high-dimensional data).

Analogously, our attacker only observes experiments (i.e., documents) but not the probabilities themselves. She can estimate the co-probabilities p^{atk}_{ij} (resp. p^{ind}_{ij}) by computing the co-frequencies $\frac{C^{\text{atk}}_{ij}}{n_{\text{atk}}} = \widehat{p}^{\text{atk}}_{ij}$ (resp. $\frac{C^{\text{ind}}_{ij}}{n_{\text{ind}}} = \widehat{p}^{\text{ind}}_{ij}$)[4]. The co-frequency is the *maximum likelihood estimator* of p_{ij}. The maximum likelihood estimator is an efficient estimator[5] for p_{ij} and converges toward this (unknown) value. The larger D_{atk} (resp. D_{ind}) is, the more precise the estimation will be. Finally, like for survey samples, the co-occurrence matrices can be obtained from different distributions. We have a particularly complex probability distribution (i.e., multivariate random variables with dependent components), so having close estimations without close distributions is unlikely.

Statistical hardness. This difficult estimation problem may serve as a security guarantee. While in classic encryption schemes, the complexity of breaking an encryption key serves as a security guarantee. The unlikelihood of finding a similar dataset could be seen as an analogous security guarantee for encrypted search. Instead of an algorithmic problem, we could base the SSE security on the complexity of a statistical problem. Hence, we are more or less in front of a **statistical hardness assumption** (opposed to the computational hardness assumptions common in cryptography).

This statistical hardness also makes sense as the attacks against SSE are not computationally hard. Hence, the security of an SSE scheme is not guaranteed because an attacker has a bounded computational power. The security of

[4]The notation $\widehat{x}$ refers to the estimation of the unknown value x.
[5]In statistics, the efficiency [16] measures the quality of an estimator. Here, the maximum likelihood estimator reaches the Cramér-Rao lower bound [10, 47] which bounds the variance of an unbiased estimator.

an SSE scheme should be guaranteed because an attacker has a knowledge of "bounded quality". This quality bound corresponds to the convergence of the statistical estimators composing the attacker knowledge. In parallel with this statistical hardness, the security of SSE also leverage computational hardness assumptions as SSE schemes rely on classic cryptographic primitives.

3.3 Document set sizes and attacker's knowledge quality

A key property of statistical estimators is the convergence when the sample size tends to infinity. The sample size has then a direct impact on the estimation quality. In our case, the sample size corresponds to the number of documents in the set. This subsection investigates *analytically and experimentally* how the document set sizes influence the attacker's knowledge quality. To measure the knowledge quality, we rely on the ϵ-similarity.

Equality of co-probabilities To simplify our analysis, we make the following assumptions: $\forall i, j \in [m], p_{ij}^{\text{ind}} = p_{ij}^{\text{atk}}$. In other words, for all pairs of keywords w_i and w_j from $\mathcal{W}$, the probability of their joint appearance in D_{atk} is equal to the probability of the same event in D_{ind}[6].

As shown in Appendix A.2, this assumption is the most advantageous setup for an attacker because it induces smaller ϵ-similarity (i.e., more similar document sets). Since our security conclusions hold for the most advantaged attacker, they hold for any attacker.

Analysis of the ϵ-similarity Let us start from the co-probability estimators (i.e., $\widehat{p}_{ij}^{\text{ind}} = \frac{C_{ij}^{\text{ind}}}{n_{\text{ind}}}$ and $\widehat{p}_{ij}^{\text{atk}} = \frac{C_{ij}^{\text{atk}}}{n_{\text{atk}}}$) which are central in the attacks. Since they are maximum likelihood estimators, we have asymptotic normality of estimators ([18] Chapter 36, Theorem 3.3). Considering n_{atk} and n_{ind} are sufficiently large, we can approximate the distribution at finite distance by the normal distribution. Under the equality of co-probabilities, we have:

$$\widehat{p}_{ij}^{\text{ind}} = p_{ij} + \frac{1}{\sqrt{n_{\text{ind}}}} Z_{ij}^{\text{ind}}, Z_{ij}^{\text{ind}} \sim \mathcal{N}(0, \sigma_{\text{ind},ij}^2), \text{ and } \widehat{p}_{ij}^{\text{atk}} = p_{ij} + \frac{1}{\sqrt{n_{\text{atk}}}} Z_{ij}^{\text{atk}}, Z_{ij}^{\text{atk}} \sim \mathcal{N}(0, \sigma_{\text{atk},ij}^2) \tag{2}$$

It implies that $(\widehat{p}_{ij}^{\text{ind}} - \widehat{p}_{ij}^{\text{atk}}) = \sqrt{\frac{1}{n_{\text{ind}}} + \frac{1}{n_{\text{atk}}}} Z_{ij}, Z_{ij} \sim \mathcal{N}(0, \sigma_{ij}^2)$. We can reuse this result in the expression of ϵ:

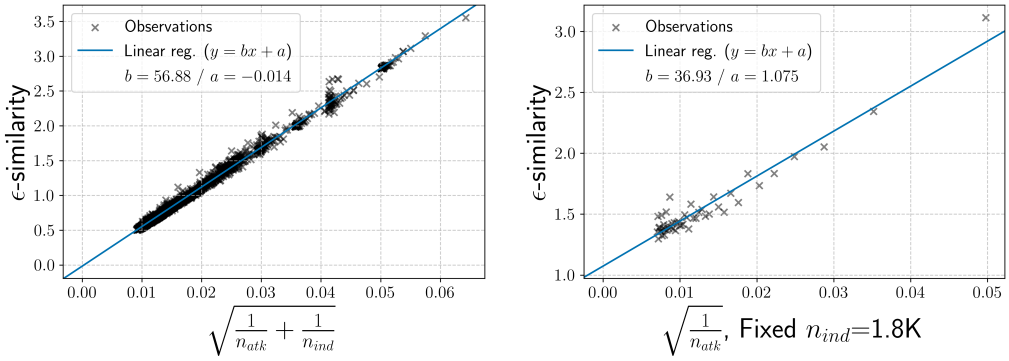
$$\epsilon = \|\text{SimMat}\| = \sqrt{\sum_{i,j} (\frac{C_{ij}^{\text{ind}}}{n_{\text{ind}}} - \frac{C_{ij}^{\text{atk}}}{n_{\text{atk}}})^2} = \sqrt{\sum_{i,j} (\widehat{p}_{ij}^{\text{ind}} - \widehat{p}_{ij}^{\text{atk}})^2} = \sqrt{\sum_{i,j} (\frac{1}{n_{\text{ind}}} + \frac{1}{n_{\text{atk}}})(Z_{ij})^2} \tag{3}$$

$$\implies \epsilon = \sqrt{(\frac{1}{n_{\text{ind}}} + \frac{1}{n_{\text{atk}}})} \sqrt{K}, K \text{ a random variable} \tag{4}$$

We can draw two conclusions from Equation (4). First, the size of the attacker document set matters as much as the size of the indexed document set. It seems evident that the more documents the attacker knows, the better the attack is. However, thinking that a bigger indexed document set helps the attacker could seem counterintuitive. To understand this intuition, we must remember that the attacks rely on comparing two estimators (i.e., the co-occurrence matrices). The attack is unsuccessful if *any of them* is poorly estimated.

Second, fixing a document set size creates a threshold for similarity. For example, let us fix n_{ind} and observe the effect on Equation (4): we have $\lim_{n_{\text{atk}} \to \infty} \epsilon = \sqrt{\frac{1}{n_{\text{ind}}} K}$. Hence, even with an infinite-sized attacker document set, the probability of having an ϵ-similarity below a certain threshold (proportional to $\sqrt{\frac{1}{n_{\text{ind}}}}$) is negligible.

[6] Assuming the equality of co-probabilities does not imply that the document set distributions are entirely equal (i.e., $\mathcal{X}^{\text{ind}} \sim \mathcal{X}^{\text{atk}}$) because it only fixes $\frac{m(m+1)}{2}$ parameters out of the 2^{m-1} parameters of these random vectors.



(a) Varying n_{atk} and n_{ind}

(b) Varying n_{atk}, Fixed n_{ind}

Fig. 1. Influence of the document set size on the similarity (Dataset: Apache)

Experimental confirmation. Figure 1a illustrates this theoretical result. To generate attack results over a wide range of similarities, we use attacker and indexed document sets, whose sizes vary between 5% and 95% of the Apache dataset. This plot comprises 2500 points with varying n_{ind} and n_{atk}. On the one hand, the linear relationship between $\sqrt{\frac{1}{n_{\text{ind}}} + \frac{1}{n_{\text{atk}}}}$ and ϵ confirms the symmetric influence of these variables. On the other hand, Figure 1b shows the similarity threshold created when n_{ind} is fixed. Indeed, the linear relationship in Figure 1a has an intercept tending to zero, while the linear relationship in Figure 1b has an intercept equal to 1. This non-zero intercept highlights the existence of a similarity threshold since, even with an infinite-sized attacker document set, the average ϵ-similarity would be 1. This threshold is *specific to the document set* (i.e., Apache), so each would have a different threshold depending on the document distribution.

3.4 Limits of distribution-independent approaches

In cryptographic schemes, security guarantees traditionally hold for any data distribution. Such distribution-independent guarantees are precious because they ensure security even in the worst case scenarios. Contrary to other cryptographic schemes, data distribution has a major influence on the attack success in SSE. Hence, distribution-independent approaches could lead to extremely conservative results due to edge-case distributions. This subsection shows why distribution-independent approaches in SSE security lead to uninformative results. To do so, we propose the following (almost) obvious theorem linking document set sizes and attack accuracy:

Theorem 1. *For any document set distribution $\mathcal{D}$ (with D_{ind} and D_{atk} drawn from $\mathcal{D}$), for any keyword universe $\mathcal{W}$, there exists a maximum index size $n_{max} \in \mathbb{N}$ such that:*

For any $n_{atk} \in \mathbb{N}$, if the number of indexed documents $n_{ind} \leq n_{max}$, the average accuracy of any passive query-recovery attack is lower than or equal to $\frac{1}{m}$.

Proof. We show that with $n_{\text{max}} = 0$, the statement in the theorem is always satisfied: the index contains no document, so all the queries return an empty set. Hence, an attacker can only randomly guess the queried keyword. The average accuracy is then $\frac{1}{m}$ (with m the number of keywords in $\mathcal{W}$). □

Theorem 1 shows that the size of the indexed document set induces a bound on the attack accuracy. The intuition behind this theorem is simple: since the document set sizes influence the knowledge quality and the knowledge quality influences the attack accuracy, the document set sizes then influence the attack accuracy.

Our proof relies on a trivial case (i.e., $n_{\max} = 0$). Many data distributions could verify the theorem with a strictly positive $n_{\max}$. However, some distributions cannot. Let us consider the keyword universe $\mathcal{W} = \{w_1, w_2\}$ and a distribution $\mathcal{D}$ such that $\mathbb{P}_{\mathcal{D}}(w_1) = 1$ and $\mathbb{P}_{\mathcal{D}}(w_2) = 0$. If D_{ind} contains even a single document, the query leakage would be sufficient to recover the queries perfectly: if the result is empty, it is w_2; otherwise, it is w_1. Moreover, there exist other distributions for which Theorem 1 holds for all $n_{\max} \in \mathbb{N}$. For example, if each indexed document contains all the keywords from $\mathcal{W}$, each query returns all the documents, so the leakage is useless (i.e., no attack is better than a random guess). Even with an infinite-sized indexed document set, the best average attack accuracy would be $\frac{1}{m}$. In practice, real-world use cases would have an $n_{\max}$ between these two edge cases.

In the previous paragraph, the first data distribution (i.e., for which the attacker has perfect accuracy) highlights the problem with distribution-independent approach. Any distribution-independent result should also hold for this specific data distribution. This would lead to uninformative/trivial results: the scheme is perfectly unsecure and we must rely on expensive attack countermeasures to ensure security. These countermeasures induce overhead that can make the SSE schemes impractical. Moreover, we also highlight a data distribution for which we have perfect security; even with no attack countermeasure. The gap between these two distribution (i.e., one with no security and one with perfect security) shows the major influence of data distributions in SSE security. We argue data-dependent approaches are preferable to obtain meaningful security assessments.

Finally, Theorem 1 emphasizes that limiting the index size can be a strong countermeasure since it guarantees a bounded accuracy. This limit depends on the keyword universe $\mathcal{W}$ and the document set distribution $\mathcal{D}$. Section 4 leverages this result to build a risk assessment protocol useful to practitioners willing to control the risk of real-world SSE deployments. The risk assessment enables estimating a non-trivial $n_{\max}$ (i.e., $n_{\max} > 0$) for specific use cases.

3.5 How to generate a realistic attacker document set?

Theorem 1 highlighted the impact of the document set distribution on attack accuracy. An attack can be dangerous or harmless, depending on the document distribution. This observation implies that attack papers must carefully choose their document set generation to provide realistic attack results. The subsection discusses how attack papers should generate the document sets.

Document set generation. Attack papers generate indexed and attacker document sets by splitting a real-world dataset (e.g., Apache emails) into two disjoint sets. By default, all papers split the dataset *uniformly at random*. We propose to compare this splitting method to a more realistic alternative: the temporal split. The temporal split consists in splitting the document set based on the document timestamp. With email datasets such as Apache, the attacker knows all the emails before a given date, and the index contains all emails sent after this date. This method simulates the recurrent motivation for the attacker's knowledge in SSE papers: data breaches. Indeed, the attacker would have access to all data before the breach date.

Accuracy discrepancy. Table 2a presents the accuracy (of the refined score attack [12]) and the ϵ-similarity obtained using the temporal split with four different splitting dates. Table 2b presents the accuracy and the ϵ-similarity obtained over 100 repetitions of the uniform split. With the temporal split, the attack accuracy is always below 5%, while it is always above 94% with the uniform split. The ϵ-similarity results confirm this observation since the ϵ is much higher

Table 2. Accuracy of the refined score attack [12] and ϵ-similarity on the Apache dataset (with $|\mathcal{W}| = 1K$)

Year	2003	2005	2007	2009
ϵ-similarity	4.90	4.07	4.52	4.62
Attack acc. (%)	0.70	2.81	2.81	1.75

(a) Temporal split

	Average	Min.	Max.
ϵ-similarity	0.62	0.59	0.71
Attack acc. (%)	98.20	94.39	99.65

(b) Uniformly random split (100 repetitions).

with the temporal split. The low accuracy with the temporal split is not due to the document set size since the split on the year 2007 provides sizes equivalent to those used for the uniform split (i.e., around 25K documents per document set). Therefore, the uniform split used in all attack papers can simulate an overly powerful attacker compared to a more realistic method.

The temporal split decreases the similarity (and the accuracy) compared to the uniform split because the distribution of the Apache dataset shifts over time. Appendix B rigorously identifies the origin of this discrepancy using statistical tests. Our observations regarding the temporal split do not hold for all possible datasets. If a data stream is not subject to a distribution shift over time, the uniform and the temporal split should provide the same results.

However, this distribution shift (also referred to "concept shift" in the literature) is common in many data-processing applications, especially in machine learning [52, 60]. Hence, this phenomenon could hinder attackers relying on data breaches from building similar document sets.

Concluding remarks. Damie et al. [12] had shown that Enron could not be used as a similar dataset to attack successfully (with their score attack) an encrypted indexed storing with the Apache dataset. Our experiments on the temporal split go one step further by showing that a subset of Apache can be insufficiently similar to a disjoint subset of Apache (due to a distribution shift).

Data breaches can provide insufficiently similar data to attack an encrypted index successfully. This observation weakens the motivation of many attack papers [1, 12, 26] justifying the existence of a similar dataset thanks to possible data breaches. Hence, finding a similar dataset is a complex task in practice. The existence of a "similar enough" attacker document set is then a strong assumption.

Our experiments also showed that the uniform split (used in attack papers) simulates a powerful attacker compared to a more realistic temporal split. However, we argue researchers must keep using a uniform split for attack analysis. On the one hand, it provides a conservative attack analysis. Indeed, a real-world attacker would not have better conditions since the uniform splitting simulates a best-case scenario for the attacker (as detailed in Appendix A). On the other hand, uniform split enables generating many document sets to repeat an attack simulation and obtain precise experimental results. Temporal splitting can only provide one dataset partitioning for a given date. It is then complex to repeat an experiment to obtain precise average results. Hence, we should use the uniform split in attack analysis but keep in mind its properties when making claims about attack practicality.

4 STATISTICAL RISK ASSESSMENT

This section shows how to assess the risk for a real-world SSE deployment. Subsection 4.1 describes a simple protocol to estimate an accuracy upper bound *specific* to a use case and discusses how to exploit this upper bound. Subsection 4.2 executes this protocol on the Enron dataset and shows what kind of conclusions a practitioner can draw. Concretely, we

demonstrate how to estimate a maximum index size guaranteeing the attack accuracy remains negligible. Subsection 4.3 shows how to use our protocol during other steps of the deployment process, especially to tune attack countermeasures.

The gap in the literature. To highlight the current gap in the literature, consider the following deployment problem: a company wants to deploy encrypted mailboxes for its employees. Each employee would have her own storage space encrypted (i.e., one SSE index per employee mailbox). The company wants to use SSE so employees can search efficiently in their encrypted emails. The system administrator wants to deploy a state-of-the-art SSE scheme but has no clue whether it is risky considering the recent attack papers [12, 21, 43].

A naive solution would be to consider Enron and Apache datasets as representative of all email use cases and study the experiment results obtained on these research datasets. Damie et al. [12] already showed that Enron is not similar to Apache in our attack context. Thus, these datasets cannot represent all email use cases.

An alternative solution would be to assume that the system administrator has a dedicated sample dataset for attack simulations. This approach has one main limitation: the size of the sample document set. For example, the sample document set can be smaller than the expected index size. In other words, its size would prevent testing realistic deployment parameters. More generally, relying only on attack simulations does not provide information about the attack accuracy with extremely large attacker knowledge. Hence, rigorous extrapolation techniques are necessary to exploit such a sample dataset efficiently.

Towards an empirical bound. We want an upper bound on the attack accuracy. Ideally, one may want to obtain a theoretical bound from mathematical analysis. Such an approach has two significant limitations. First, the attack problem is complex, which makes the analysis challenging. Islam et al. [26] proved the attack problem is NP-complete. Second, a theoretical bound could be non-informative. Subsection 3.4 already highlighted the weakness of distribution-independent approaches. Hence, a purely theoretical bound could be too loose due to the edge cases.

To avoid these issues, we propose to rely on empirical bounds. This empirical approach would consider the specificities of a use case (**represented by a sample dataset**) to obtain tight bounds. We can use statistical tools to estimate an upper bound with high probability. This upper bound should not be absolute but simply statistical: it is not impossible to be above this threshold; it is only unlikely. A successful attack is always possible, but this chance must be as small as possible. Finally, we can also leverage this bound to deduce the maximum index size $n_{\max}$ introduced in the Theorem 1.

4.1 Estimating the risk

We call *risk* the maximum attack accuracy with probability α. We want to estimate this upper bound using simulation results (obtained on a representative sample dataset).

Subsections 3.3 and 3.4 highlighted the central role of the document set sizes in the attack success. Hence, we want to estimate our bound in function of n_{ind} and n_{atk}. Then, we fix the rest of the simulation parameters to ensure a *conservative* risk assessment; real-world attackers cannot benefit from better conditions . Thus, our simulations rely on four assumptions: *(1)* the attacker knows the whole keyword universe, *(2)* the attacker observed all possible queries, *(3)* the attacker knows k known queries, *(4)* we use uniform splitting for document set generation.

The first two assumptions are simple: we assume maximum attacker knowledge. Then, we assume a limited number of known queries; otherwise, there are no unknown queries to attack. This threshold should correspond to the number of queries under which a (passive or active) attack is considered unsuccessful by the practitioner. Finally, Appendix A details why uniform splitting simulates a best-case scenario for the attacker. These assumptions can give the impression that we consider an overly powerful adversary. However, these assumptions are standard in the attack literature. A side

contribution of our paper is also to highlight the unexpected strength of these assumptions, especially the uniform splitting analyzed in Section 3.5.

Quantile regression. Our bound estimation is based on quantile regression [23, 32, 33]. In a quantile regression, the resulting estimated function describes the quantile of a data distribution[7] instead of the average case (as linear regression does). A quantile regression computes the parameters (b, a) such that $Q_Y(\alpha) = b \cdot X + a$, for (X, Y) two data distributions and α a quantile level. We refer to [23, 32, 33] for details about the computation. This quantile regression is ideal for representing a maximum accuracy bound with high probability.

Estimation protocol. We want to estimate the quantile function describing the quantile α of accuracy for a given attack, with the document sizes as input parameters: $Q_{\mathrm{Acc}}(\alpha; n_{\mathrm{ind}}, n_{\mathrm{atk}})$. Section 5 presents a similar estimation protocol to estimate the average accuracy. The current subsection presents briefly the intuitions behind our estimation protocol, but a reader can refer to Section 5 for a more incremental approach with supporting plots.

We cannot run the quantile regression on the raw simulation results. The quantile regression outputs an affine function, so we must compute this regression in a space where our variables have a linear relationship. The raw variables cannot have a linear relationship because the accuracy is in $[0, 1]$, and the document sizes are in $\mathbb{N}$. To solve this kind of regression problem, the logit function (and its inverse expit) is traditionally used in statistics[8]. The logit function maps the space $(0, 1)$ onto $\mathbb{R}$. With this logarithmic transformation, having a linear relationship is possible since we have a variable in $\mathbb{R}$ (i.e., logit(Acc)) and two variables in $\mathbb{N}$ (i.e., the document set sizes). In practice, we process the document set sizes as real numbers.

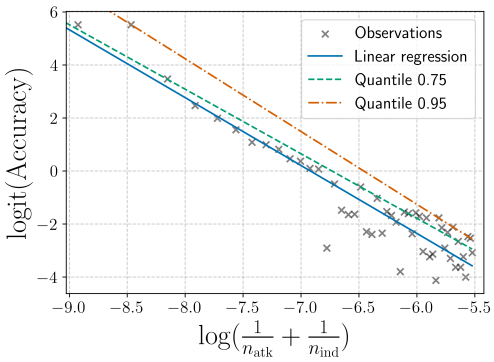
However, this logarithmic transformation is not enough to have a linear function to estimate. Subsection 3.3 highlighted the linear relationship between $\sqrt{\frac{1}{n_{\mathrm{ind}}} + \frac{1}{n_{\mathrm{atk}}}}$ and the similarity metric ϵ. Hence, we take inspiration from this linear relationship to solve our current quantile regression problem. We end up computing the quantile regression between logit(Acc) and $\log(\frac{1}{n_{\mathrm{ind}}} + \frac{1}{n_{\mathrm{atk}}})$. To sum up, our quantile regression (at level $\alpha \in (0, 1)$) outputs the parameter a and b such that $Q_{\mathrm{logit(Acc)}}(\alpha; n_{\mathrm{ind}}, n_{\mathrm{atk}}) = b \log(\frac{1}{n_{\mathrm{ind}}} + \frac{1}{n_{\mathrm{atk}}}) + a$. Once we have the parameters a and b, we deduce the accuracy upper bound: $Q_{\mathrm{Acc}}(\alpha; n_{\mathrm{ind}}, n_{\mathrm{atk}}) = \mathrm{expit}(b \log(\frac{1}{n_{\mathrm{ind}}} + \frac{1}{n_{\mathrm{atk}}}) + a)$.

Now that we have a satisfying estimation protocol, we only need to generate simulation results (using a sample dataset representative of the use case) with varying document set sizes and compute the quantile regression. Figure 11 of Appendix D details our simulation loop. This algorithm is straightforward, and it leverages the symmetrical influence of n_{atk} and n_{ind} (highlighted in Subsection 3.3) to optimize the simulation process.

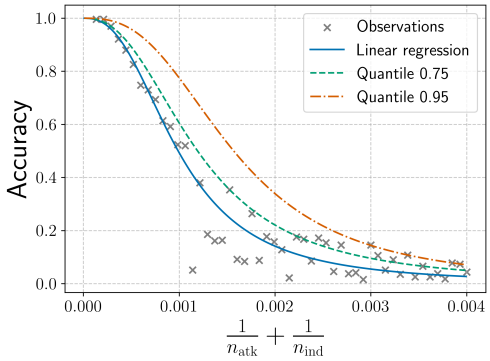
Exploiting the regression model. Using this estimated function, we can compute the maximum index size introduced in Theorem 1: $n_{\max}$ is such that $\lim_{n_{\mathrm{atk}} \to \infty} Q_{\mathrm{Acc}}(\alpha; n_{\max}, n_{\mathrm{atk}}) \leq \frac{1}{m}$. We can generalize this formula and consider a generic maximum accuracy threshold $\beta_{\max}$ (in $[0, 1]$) instead of $\frac{1}{m}$. To obtain a higher $n_{\max}$, we can also relax an assumption by bounding the attacker's document set size.

Defining SSE security requirements. We can define SSE security requirements using a pair $(\beta_{\max}, n_{\max})$, with $\beta_{\max}$ the maximum attack accuracy and $n_{\max}$ the maximum index size. On the hand, $\beta_{\max}$ should be a commonly agreed threshold (e.g., 5%). On the other hand, $\beta_{\max}$ should be set based on the use case properties. Our risk assessment enables estimating pairs guaranteeing security for a given use case. A use case is insecure if no pair satisfies the practitioners

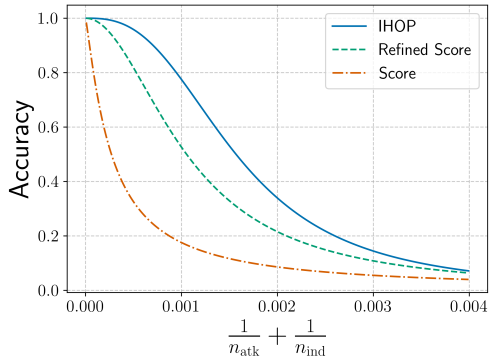
[7] The quantile of level α for distribution Y is defined as follows: $Q_Y(\alpha) = \inf\{y : F_Y(y) \geq \alpha\}$, with F_Y the cumulative distribution function of Y.
[8] $\mathrm{logit}(p) = \log(\frac{p}{1-p})$ with $p \in (0, 1)$, with a known inverse function expit.



(a) With the logarithmic transformation (IHOP attack [43])



(b) Estimated function mapped in the initial space (IHOP attack [43])



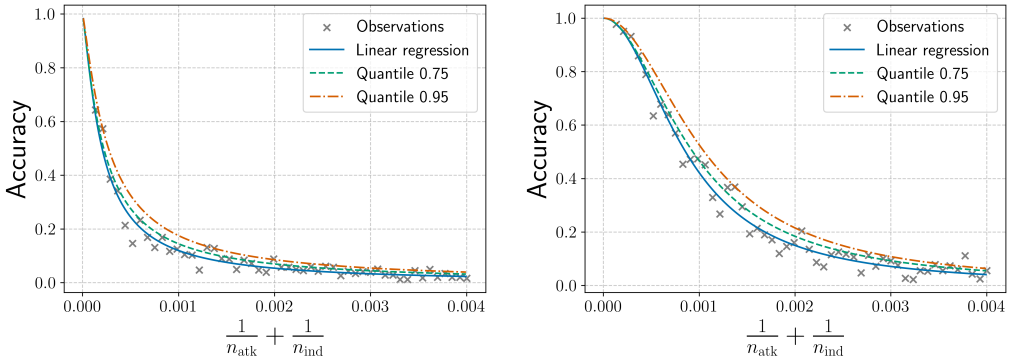
(c) Comparison of estimated upper bounds of three attacks [12, 43] with the quantile 0.95

Fig. 2. Estimated upper bounds (Keyword universe size: 500 / Dataset: Enron)

expectations (e.g., the maximum index size is too low). This maximum index size is then a precious tool for practitioners to *deploy SSE with security guarantees*.

Divergence from Theorem 1. Our estimated $n_{\max}$ slightly differs from Theorem 1. On the one hand, our risk assessment protocol studies a special case of the theorem where D_{sim} and D_{atk} are drawn independently because we focus on similar-data attacks. On the other hand, we do not consider the average accuracy contrary to the theorem. Our bound holds for the attack accuracy in general *with high probability*.

Theorem 1 focuses on the average case differs because it has a shorter formulation. However, we could prove a variant providing an upper bound with high probability using a similar proof. To understand the difference between the "average" and "high probability" cases, we can see the accuracy results as a Gaussian distribution. The average case fixes the threshold at the middle of the bell curve, while the high probability case fixes the threshold "far enough on the right" so only a negligible proportion of points are above the threshold. Hence, both cases rely on the same intuitions about the staistical nature of the attacker's knowledge.



(a) Score attack

(b) Refined Score attack

Fig. 3. Estimated upper bounds for different attacks [12, 43] (Keyword universe size: 500 / Dataset: Enron)

4.2 Concrete example

This subsection performs our risk assessment on the example given at the beginning of the section (i.e., deployment of SSE for company mailboxes). This example uses the Enron dataset as a sample dataset. Hence, the conclusions only hold for the Enron company. Any other company must gather a representative dataset of its internal emails and execute the risk assessment protocol on this sample. This example only aims to demonstrate the potential of our risk assessment.

Upper bound estimation. The company has a keyword universe composed of the 500 most frequent keywords. The system administrator considers that discovering more than 15 queries would already be a successful attack, so we set the number of 15 known queries[9]. Figure 2 shows the estimation result for the IHOP attack. We observe a smooth and coherent upper bound (with regard to the simulation results). Figure 3 provides equivalent results for two other attacks: score and refined score attacks [12]. We plot the 0.95 quantiles of all three attacks in Figure 2c. Figure 2c shows that IHOP outperforms the refined score attack so we use IHOP as our reference upper bound for risk assessment.

Maximum index size estimation. Taking inspiration from Theorem 1, we can use the upper bound to deduce a maximum index size. Figure 4 shows the maximum index size for varying maximum accuracies. The top curve (in blue) depicts the most conservative setup: adversary with infinite-sized similar document set. For example, a maximum accuracy of 5% requires a maximum size of 218 documents per index. This constraint would contribute to keeping small indices (e.g., via mandatory regular cleaning of mailboxes) or subdividing indices (e.g., one per email folder). However, 200 might be too limiting for our example use case.

A solution to this problem would be to relax our assumptions and consider an attacker with finite-sized document set. The three bottom curves of Figure 4 shows the estimation with three different bounds on the attacker document set size: 1000, 500, 200. These curves show a clear decrease of the maximum attack accuracy.

Feasability. Finally, we must explain whether our hypothetical system administrator with limited knowledge of cryptography can execute it. Let us skim through the parameters to explain why a system administrator can set them. In Figure 11 of Appendix D, the simulation algorithm takes as input some "fixed parameters." They correspond to the

[9]Most experiments in [12] assumed between 10 and 20 known queries.

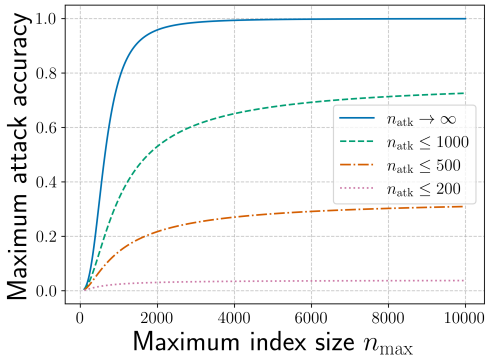


Fig. 4. Maximum index size for varying maximum accuracies and maximum attacker document set sizes (quantile regression with $\alpha = 0.95$ / Dataset: Enron / Attack: IHOP [43])

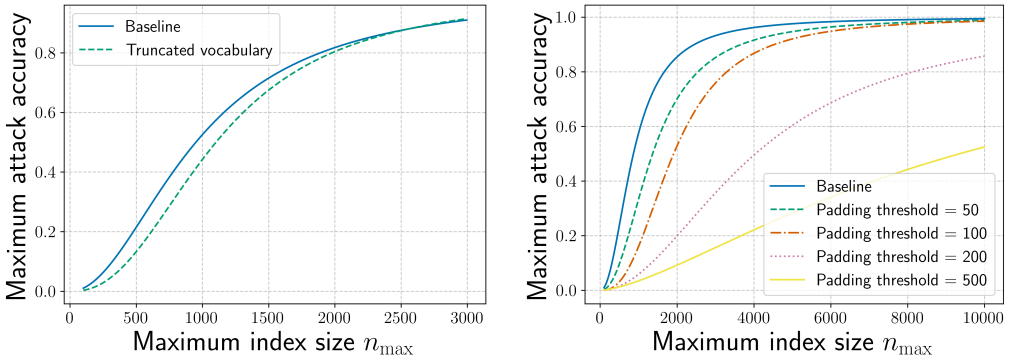
deployment parameters (e.g., keyword universe size) that must be set by the system administrator even if she deploys SSE without a prior risk assessment. Moreover, one can set the number of experiments depending on the time available. The more experiments one performs, the more accurate the quantile regression will be.

We have two non-trivial parameters left: the attack and the quantile. We argue that these two parameters should be set with the help of the research community. We would refer to software such as LEAKER [27] to provide up-to-date attack lists. This risk assessment is comparable to concrete security approaches where the security is re-estimated every time a new attack is released. We recommend using the thresholds 0.1%, 1%, and 5%, recurrent in statistics for the quantile parameter. The 5% threshold could be sufficient because our simulation parameters are so conservative (e.g., uniform splitting) that a realistic attacker should not approach it.

4.3 Strengthening SSE deployments

Suppose the system administrator is unsatisfied with the previously estimated security guarantees (i.e., the maximum index size is too small). We can use our risk assessment to choose the best index parameters and to tune countermeasures (also referred to as "attack mitigations" in some papers). These additional steps strengthen the security and enable deployment with a satisfying security guarantees.

Choosing the best parameters. The first strengthening strategy is to choose adequate parameters for the index. As shown in [12], some index parameters (e.g., the distribution or the size of the keyword universe) influence the attacker's success. The risk assessment can help choose the parameters minimizing the attack success. To test this idea, we focus on the keyword universe. In Section 4, we use the top 500 most frequent keywords as keyword universe, but it could be a poor choice from a security point of view. The most frequent keywords have a more distinguishable distribution, hence, are easier to recover [12]. Comparing the estimated function obtained using different keyword universes is then a solution to make a wise decision. Figure 5a compares the maximum accuracy bounds estimated respectively with our baseline (i.e., the most frequent 500 keywords) and with a truncated keyword universe (i.e., we keep from the 101st most frequent keyword to the 500th most frequent keyword). It shows an apparent decrease in accuracy when the vocabulary is truncated. This approach can be generalized to any "tuneable" index parameter.



(a) With two different keyword universes.

(b) With varying countermeasure parameters [6]

Fig. 5. Maximum accuracy estimation applied to parameter tuning (Attack: Refined Score attack [12] / Conservative estimation: $n_{\text{atk}} \longrightarrow \infty$ / Dataset: Enron)

How much mitigation is needed? A second problem related to the parameter choice is the countermeasure choice. In the literature, several countermeasures have been proposed [6, 13, 19, 56]. These solutions reduce the attack accuracy in exchange for some overhead. However, it is unclear whether they are needed in practice and, if so, how much mitigation is needed. To answer the first part of the question, we can run our risk assessment without attack mitigation, and if the risk is too high, a countermeasure is needed. Once the need is established, we need to tune the countermeasure, and the statistical risk assessment is again helpful.

The countermeasure tuning is similar to the parameter tuning: we compare maximum attack accuracy with varying parameters. The goal is to find the countermeasure parameter minimizing the overhead while satisfying the security expectations of the administrator. We focus on the straightforward padding countermeasure proposed in [6]: it randomly adds false positive results to the index so that all the queries return a number of documents multiple of t, a padding threshold. The bigger t is, the more effective the countermeasure and the more overhead there is. Figure 5b shows the maximum accuracy bounds (in function of the maximum index size) with varying padding thresholds. We observe that the maximum accuracy decreases while the padding threshold increases. A system administrator can leverage this observation to find the minimum padding threshold t fulfilling its security expectations, especially to reach an acceptable maximum index size.

Besides parameter tuning, we can also compare several countermeasures to choose the most efficient technique for a given use case. The approach would be identical to the parameter tuning: estimate and compare upper bounds.

5 ATTACK ANALYSIS BASED ON A SIMILARITY METRIC

Attack papers usually represent the attack accuracy for varying document set sizes. As highlighted in Subsection 3.3, these parameters directly impact the quality of the attacker's knowledge. However, they do not measure knowledge quality contrary to similarity metrics. Hence, observing the attack accuracy with varying similarity would provide more meaningful figures representing the link between knowledge quality and attack success.

Moreover, the existing attack papers often induce a bias when they fix one of the document set sizes (usually n_{ind}). Indeed, Subsection 3.3 also showed that a fixed document set size creates a threshold on the attack accuracy. To avoid

this phenomenon, one may try to represent the attack accuracy with varying n_{ind} **and** n_{atk}, which would lead to poorly readable results. Basing an attack analysis on a similarity metric solves this issue, because there are only two dimensions to observe: similarity and accuracy.

This section provides tools and results to improve the attack analysis methodology using a similarity metric. As in Section 3, we rely on ϵ-similarity. This section answers three open questions:

- Subsection 5.1: How to analyze attack success using a similarity metric?
- Subsection 5.2: Is similarity the only parameter influencing the accuracy of an attack?
- Subsection 5.3: How can similarity metrics improve attack comparison?

5.1 How to analyze attack success using a similarity metric?

This subsection analyzes the accuracy of an attack by estimating its average accuracy function. As motivated previously, we propose to use ϵ-similarity instead of document set sizes to represent our results. In other words, we want to estimate a function $\widehat{f}_{\text{Acc}}$ such that $\widehat{f}_{\text{Acc}}(\epsilon) = \mathbb{E}(\text{Acc})$[10], we are in front of a regression problem. A continuous function provides a more detailed understanding of the attack's strengths and weaknesses. Such functions enable extrapolating the simulation results and precisely identify gaps in the literature. Attack papers usually represent the accuracy on a finite set of points, which gives a poor understanding of the complete attack behavior. Subsections 5.2 and 5.3 will show how we can use these functions to provide novel insights about the existing attacks.

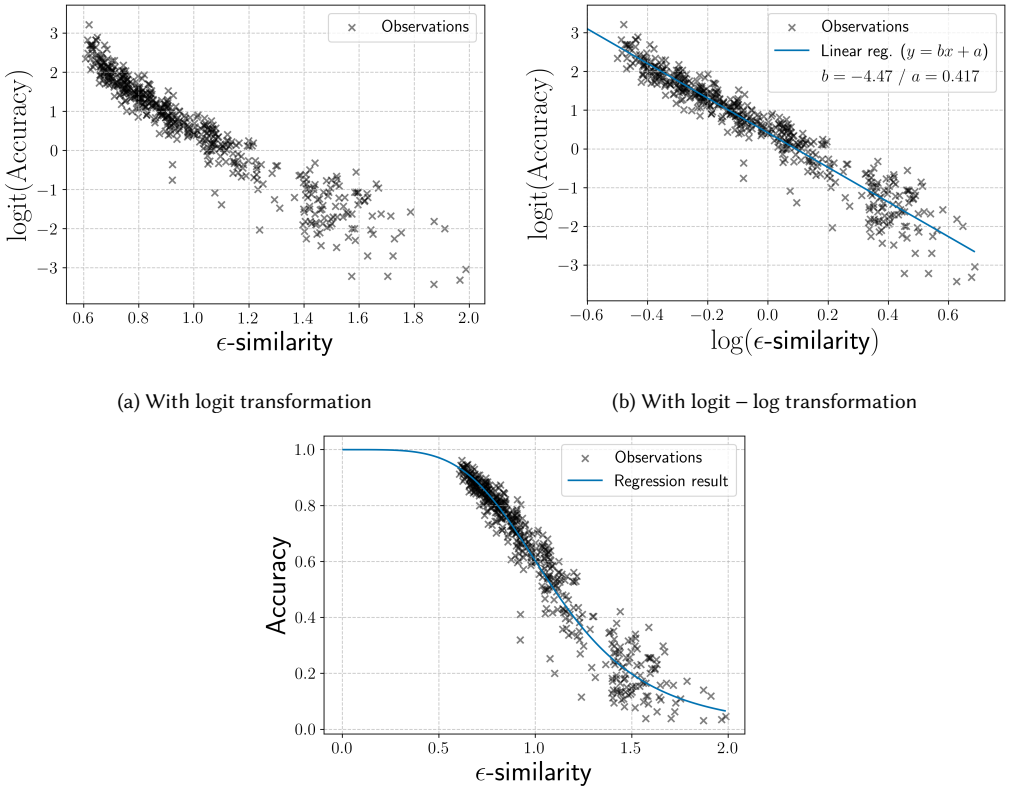
Estimating the accuracy function. First, we can point out that the function cannot be linear since the accuracy is in $[0, 1]$. Hence, we cannot directly use linear regression on the raw data. To circumvent this difficulty, we estimate a function $\widehat{f}'_{\text{Acc}}(\epsilon) = \text{logit}(\text{Acc})$. The logit function is defined as $\text{logit}(p) = \log(\frac{p}{1-p})$ with $p \in (0, 1)$ and has a known inverse function expit. This logit function maps a $(0, 1)$ space into the real number space. This logarithmic transformation is common in statistics.

Figure 6a represents the simulation results with the logit transformation. We observe two problems in this figure: we do not have apparent linearity between the variables, and there is a "heteroscedasticity of the noises". The first problem is simple: the distribution of the points seems flattened when ϵ grows. The second issue concerns the assumption made when computing a linear regression. A fundamental assumption in linear regression is that the noise distribution is the same at each point of the space. Then, the data should have the same noise distribution for small and high ϵ, which is not the case in Figure 6a. A recurrent solution in ML to the linearity problem is to apply a logarithm transformation on the x axis. Figure 6b presents the results of a linear regression between $\text{logit}(\text{Accuracy})$ and $\log(\epsilon)$. Combining these two logarithmic transformations makes the linear relationship more apparent. Moreover, the logarithmic transformation slightly corrected the heteroscedasticity. Scaling methods (where a data point is modified by a scaling factor depending on its ϵ) exist to perfectly fix heteroscedasticity. However, they add more complexity for equivalent results.

To sum up, we map the attack simulation results in a logit – log space to compute a linear regression (i.e., $\text{logit}(\mathbb{E}(\text{Acc})) = b\log(\epsilon) + a$). We then deduce the average accuracy function $\widehat{f}_{\text{Acc}}(\epsilon) = \text{expit}(b \cdot \log(\epsilon) + a)$ with (b, a) the regression parameters. We represent the estimated function for the Enron dataset in Figure 6. We demonstrate this protocol using the refined score attack, but it can be performed with any other attack (see Subsection 5.3).

Alternative regression methods. Several theoretical considerations support our regression model, and it produces convincing results. However, we cannot formally prove that it is the best model to represent the relationship between

[10] $\mathbb{E}[X]$ is the expected value of the random variable X.



(a) With logit transformation

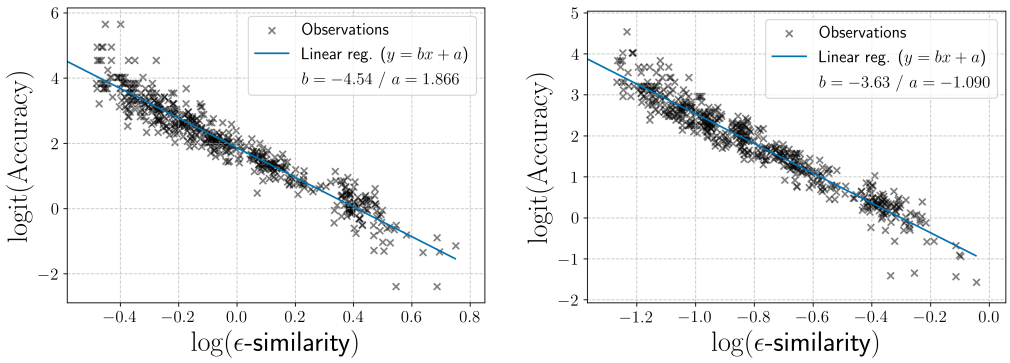
(b) With logit – log transformation

(c) In the initial space

Fig. 6. Analysis of the refined score attack on Enron dataset ($m = 1K$)

accuracy and similarity. The machine learning literature has described many other regression models, but our method is simple and provides interpretable results despite an initial non-linear problem. A model is "interpretable" if an observer can understand the cause of a decision [39] (e.g., predict the impact of an input variation). This interpretability is absent in many popular models, such as neural networks and random forests. These models can solve non-linear problems efficiently, but are often referred to as "black boxes" due to their lack of interpretability. Our model is interpretable because it combines linear models with simple logarithmic transformations. Interpretability is a crucial property in our case to foresee the security limits. The second strength of our model is its extension to risk assessment (see Section 4) via the replacement of linear regression by quantile regression.

During our study, we observed a particular behavior on the "tail results"; i.e., the results with extreme ϵ. Appendix C presents minor tweaks to deal with these extreme cases.



(a) Apache dataset

(b) Blogs dataset

Fig. 7. Regression results on others datasets ($m = 1K$)

5.2 The role of similarity in attack success

We can now use our average accuracy functions to answer the following question: is similarity the only factor influencing the attack success? We can reformulate it as follows: for each attack, is there a unique accuracy function $\widehat{f_{\text{Acc}}}(\epsilon)$ valid for all document sets?

We reproduced the average accuracy estimation for two other datasets: Apache and Blogs (see Figure 7). The linear regression parameters are $(-4.54, 1.87)$ for the Blogs dataset, $(-3.63, -1.09)$ for Apache, and $(-4.47, 0.417)$ for Enron. These parameters significantly differ, so no unique function links the similarity to the attack accuracy. Hence, similarity is not the only parameter influencing attack accuracy.

We can explain this phenomenon using keyword distribution. Consider a hypothetical dataset with m keywords with equal probabilities and co-probabilities (e.g., all keywords appear in all documents). Even with perfect knowledge of the probabilities (i.e., $\epsilon = 0$), we cannot distinguish the keywords with an accuracy better than a random guess (i.e., $\mathbb{E}(\text{Acc}) = \frac{1}{m}$). Hence, each dataset has a keyword distribution, resulting in more or less distinguishable keywords. This observation is in line with our claims about distribution-independent approaches in Subsection 3.4.

Understanding the intent of countermeasures. The dependence of the attack accuracy on keyword distribution is also interesting to put into perspective the works on countermeasures. Indistinguishability is not a new notion in SSE [4, 11, 55], but we can revisit it thanks to our results. Countermeasures usually aim at some keyword/query indistinguishability. The intuition is to have all queries leaking the same information, so the leakage induced by SSE schemes becomes useless to an attacker.

For example, the countermeasures based on the injection of false-positive results [4, 6, 56] (used in SSE implementations such as ShieldDB [53]) produces "undistinguishable" queries by "smoothing" the keyword frequencies. We argue that these countermeasures do not produce indistinguishability but simply noisy statistics. Indeed, the frequencies might be indistinguishable, but the co-frequencies do not automatically inherit this property. To produce an indistinguishable leakage, we would need to smooth all the statistics, from the keyword frequencies to the co-frequencies of m keywords (i.e., the $2^m - 1$ parameters of the random binary vectors defined in Subsection 3.2).

However, the good performances of the countermeasures show that noisy statistics are enough to prevent attacks. We do not need indistinguishable leakages; we must only ensure they are realistically unusable. For example, two queries can leak distinct co-frequency information; the attacks will be unsuccessful if this information leakage is too noisy. Hence, it is unnecessary to perfectly protect the queries by enforcing them to have the same co-frequency leakage. To reach a satisfying noise level, we suggest using our risk assessment variant for countermeasure tuning (Subsection 4.3).

We identify as an open problem the composition of our co-occurrence mathematical representation with the countermeasure mathematical model of [21]. Such an analysis may lead to improvements in attack countermeasures.

5.3 Comparison framework

Finally, we can use the average accuracy functions to improve the attack comparison methodology. Our comparison protocol can be summarized as follows: *(1)* generate simulation results for each attack, *(2)* estimate the average accuracy function of each attack, *(3)* compare the average accuracy functions.

Figure 8 compares the accuracy functions of three recent similar-data attacks (on Apache dataset): Score, Refined score, and IHOP attacks. This experiment uses the (straightforward) Algorithm 10 of Appendix 10 to generate results with varying ϵ.

The IHOP and refined score attacks outperform the score attack. IHOP seems to obtain slightly better results for smaller ϵ while for higher ϵ, the refined score attack has a small advantage. In the context of risk assessment, the IHOP risk function was always above the refined score risk function in Figure 2c. These observations mean that IHOP occasionally reaches very high accuracies, while refined score results have a smaller variance. Despite these minor differences, both attacks are accurate in similar scenarios, but IHOP is better because it requires no known queries.

This comparison framework simplifies the identification of game-changing attacks. The existing attack papers typically compare attacks over a set of parameters, emphasizing a clear accuracy difference between a novel attack and the state-of-the-art. While this approach identifies improvements, it does not give a full picture of the situation. In our case, a traditional attack comparison between IHOP and Refined Score attack would "zoom in" between $\epsilon = 0.5$ and $\epsilon = 1.0$ to focus on the most considerable improvements. This "zoom" ignores subtle phenomena, such as the slight advantage of the Refined Score attack on highly dissimilar document sets. Systematically comparing accuracy functions would guarantee attack papers thoroughly analyze the attack behaviors and avoid focusing on a convenient set of parameters. Moreover, the linear regression estimates an average distribution, smoothing the noise over the curve. Noisy experimental results are a recurrent concern that often questions whether an accuracy difference is significant. Linear regression remains a statistical estimation, so some uncertainty remains. We can model this uncertainty using confidence intervals.

As for individual experimental results, we can run simple statistical tests to prove a difference to be statistically significant. While performing statistical tests on a large set of individual experimental results is tedious, performing a statistical test on linear regression parameters [9] is much easier. In other words, we can statistically prove that an estimated function is significantly higher than another function instead of proving a significant difference on individual points. Hence, this simplicity should *encourage future attack papers to rigorously prove accuracy improvements using statistical tests on accuracy functions.*

Countermeasure efficiency. Finally, the comparison framework can be extended to countermeasure comparison. The idea would be to compare accuracy functions obtained with and without countermeasures. The impact of the

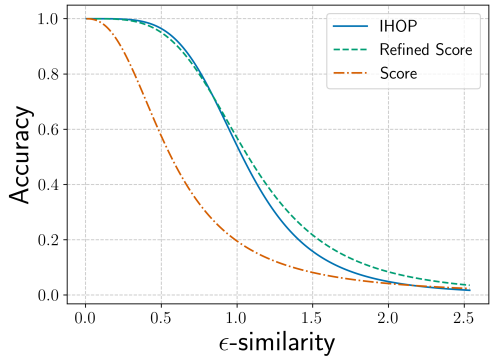


Fig. 8. Comparison of the average accuracy functions of the Score, Refined Score, and IHOP attacks ($m = 500$)

countermeasure on the accuracy function would then quantify the efficiency of a countermeasure. We do not provide experimental results, but Section 4 presents an analogous extension to countermeasures for risk assessment.

6 EXTENSIONS TO OTHER SCHEMES AND ATTACKS

Our work focuses on similar-data attacks against SSE schemes with access and search pattern leakage. This section widens our scope and discusses the extension of our results to different settings.

6.1 Extension to schemes with less leakage

Our results naturally extend to SSE schemes with less leakage. For example, we can consider with volume pattern leakage instead of access pattern leakage. While access pattern leakage reveals the list of document identifiers matching a query, the volume pattern leakage only reveals the number of document identifiers. The volume pattern leakage corresponds to the diagonal of the co-occurrence matrices described in Section 3. Hence, our analysis and results hold for these schemes. This setting simply reduces the amount of (noisy) information available to the attacker.

6.2 Extension to schemes supporting conjunctive-keyword search

Dijkslag et al. [14] described a generic technique to attack schemes supporting conjunctive-keyword search using the attacks against schemes with single-keyword search. Our results naturally extend to these schemes. The only modification is the similarity metric: instead of building a similarity matrix from the co-occurrence matrices, we build co-occurrence tensors because these attacks exploit the co-occurrences of k different keywords, and then the ϵ-similarity metric is the tensor norm. Hence, we have two arrays of statistical estimators and the attack success is conditioned by the statistical estimation quality. Our insights regarding document set sizes and data distribution still hold.

6.3 Extension to known-data attacks

The first direct extension is known-data attacks. These attacks are tightly related to similar-data attacks because only a small parameter changes: the attacker-known documents are indexed. Hence, our insights hold for known-data attacks. However, the mathematical model of Section 3 must be adapted because the attacker and indexed document sets are two dependent random variables. Some additional work is necessary to extend all our results to known-data attacks.

Road to new attacks. This extension also raises new research questions for attacks. Until now, all attack papers presented either similar-data or known-data attacks. We may see a continuum between these attacks and conceive hybrid attacks with an attacker's knowledge composed of indexed and non-indexed documents. Such attacks may reach even higher accuracies.

6.4 Extension to attacks using query frequency

Liu et al. [37] proposed an attack against SSE using query frequency. In this attack, the attacker has access to a reference dataset with keyword query frequencies and tries to match the reference frequencies to the frequency of the observed queries. Our results do not extend to these attacks; as they rely on different statistical information.

However, we can reuse our statistical approach to analyze these attacks. The query frequency is also a statistical estimator, so the attack relies once again on the comparison of two statistical estimators. Hence, the attack success will also be influenced by the statistical estimation quality. While our analysis highlighted the importance of the document set sizes, these attacks are influenced by the number of queries. Indeed, the query frequency estimation converges with the number of queries observed.

For attacks relying on access pattern, our paper set a maximum index size guaranteeing an appropriate security level. For attacks relying on query frequency, we can set a maximum number of queries: below this threshold, the adversary's knowledge is too noisy to successfully attack the encrypted index. Evertyime this number of query is reached, the user could refresh the encrypted index in order to mitigate attacks.

CONCLUSION

Our work provided a novel understanding of the attacker's knowledge in passive attacks on encrypted search. We modeled the attacker's knowledge using statistical estimators and highlighted the noise naturally contained in this knowledge. We leveraged this model to provide several novel insights about SSE attacks, especially the weakness of distribution-independent security assessment. Then, we built upon this intuition a statistical framework to assess the risk of specific real-world use cases. Finally, we improved the attack analysis methodology using analogous statistical methods. Hence, our results promoted new practices for practitioners and researchers. In particular, our risk assessment protocol supports real-world deployments with controlled risk.

These results raise new questions about the security approach for privacy-preserving technologies with information leakage, such as SSE. This leakage is problematic from a theoretical perspective because it prevents any theoretical security proof. However, the noisy nature of leakage can act as a natural attack mitigation. Statistical approaches such as ours can provide a novel form of security guarantees for these technologies.

ACKNOWLEDGMENTS

This work was partially supported by ANR project ANR-20-CE23-0013 'PMR'.

REFERENCES

[1] Laura Blackstone, Seny Kamara, and Tarik Moataz. 2020. Revisiting Leakage Abuse Attacks. In *Network and Distributed System Security Symposium (NDSS)*.

[2] J. Martin Bland and Douglas G. Altman. 1995. Multiple significance tests: the Bonferroni method. *BMJ* 310, 6973 (Jan. 1995), 170. https://doi.org/10.1136/bmj.310.6973.170 Publisher: British Medical Journal Publishing Group Section: General practice.

[3] Raphael Bost. 2016. Sophos: Forward Secure Searchable Encryption. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*. Association for Computing Machinery, 1143–1154. https://doi.org/10.1145/2976749.2978303

[4] Raphael Bost and Pierre-Alain Fouque. 2017. Thwarting Leakage Abuse Attacks against Searchable Encryption – A Formal Approach and Applications to Database Padding. https://eprint.iacr.org/2017/1060

[5] Raphaël Bost, Brice Minaud, and Olga Ohrimenko. 2017. Forward and Backward Private Searchable Encryption from Constrained Cryptographic Primitives. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1465–1482. https://doi.org/10.1145/3133956.3133980

[6] David Cash, Paul Grubbs, Jason Perry, and Thomas Ristenpart. 2015. Leakage-Abuse Attacks Against Searchable Encryption. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15)*. Association for Computing Machinery, 668–679. https://doi.org/10.1145/2810103.2813700

[7] David Cash, Joseph Jaeger, Stanislaw Jarecki, Charanjit Jutla, Hugo Krawczyk, Marcel-Cătălin Roşu, and Michael Steiner. 2014. Dynamic Searchable Encryption in Very-Large Databases: Data Structures and Implementation. In *Proceedings 2014 Network and Distributed System Security Symposium*. Internet Society. https://doi.org/10.14722/ndss.2014.23264

[8] Melissa Chase and Seny Kamara. 2010. Structured Encryption and Controlled Disclosure. In *Advances in Cryptology - ASIACRYPT 2010 (Lecture Notes in Computer Science)*, Masayuki Abe (Ed.). Springer, 577–594. https://doi.org/10.1007/978-3-642-17373-8_33

[9] Clifford C. Clogg, Eva Petkova, and Adamantios Haritou. 1995. Statistical Methods for Comparing Regression Coefficients Between Models. *Amer. J. Sociology* 100, 5 (March 1995), 1261–1293. https://doi.org/10.1086/230638

[10] Harald Cramér. 1999. *Mathematical Methods of Statistics*. Princeton University Press.

[11] Reza Curtmola, Juan Garay, Seny Kamara, and Rafail Ostrovsky. 2006. Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions. In *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)*. Association for Computing Machinery, 79–88. https://doi.org/10.1145/1180405.1180417

[12] Marc Damie, Florian Hahn, and Andreas Peter. 2021. A Highly Accurate Query-Recovery Attack against Searchable Encryption using Non-Indexed Documents. In *30th USENIX Security Symposium (USENIX Security 21)*. 143–160.

[13] Ioannis Demertzis, Dimitrios Papadopoulos, Charalampos Papamanthou, and Saurabh Shintre. 2020. SEAL: Attack Mitigation for Encrypted Databases via Adjustable Leakage. In *29th USENIX Security Symposium (USENIX Security 20)*. 2433–2450.

[14] Marco Dijkslag, Marc Damie, Florian Hahn, and Andreas Peter. 2022. Passive Query-Recovery Attack Against Secure Conjunctive Keyword Search Schemes. In *Applied Cryptography and Network Security (Lecture Notes in Computer Science)*, Giuseppe Ateniese and Daniele Venturi (Eds.). Springer International Publishing, 126–146. https://doi.org/10.1007/978-3-031-09234-3_7

[15] Dominique Dittert, Thomas Schneider, and Amos Treiber. 2023. Too Close for Comfort? Measuring Success of Sampled-Data Leakage Attacks Against Encrypted Search. In *Cloud Computing Security Workshop (CCSW '23)*.

[16] Brian S. Everitt and Anders Skrondal. 2010. *The Cambridge dictionary of statistics*. Cambridge University Press.

[17] Javad Ghareh Chamani, Dimitrios Papadopoulos, Charalampos Papamanthou, and Rasool Jalili. 2018. New Constructions for Forward and Backward Private Symmetric Searchable Encryption. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1038–1055. https://doi.org/10.1145/3243734.3243833

[18] Zvi Griliches, James J Heckman, Michael D Intriligator, Robert F Engle, Edward E Leamer, and Daniel McFadden. 1983. *Handbook of econometrics*. Elsevier.

[19] Paul Grubbs, Anurag Khandelwal, Marie-Sarah Lacharité, Lloyd Brown, Lucy Li, Rachit Agarwal, and Thomas Ristenpart. 2020. Pancake: Frequency Smoothing for Encrypted Data Stores. In *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, 2451–2468.

[20] Paul Grubbs, Marie-Sarah Lacharité, Brice Minaud, and Kenneth G Paterson. 2018. Pump up the volume: Practical database reconstruction from volume leakage on range queries. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. 315–331.

[21] Zichen Gui, Kenneth G Paterson, and Sikhar Patranabis. 2023. Rethinking Searchable Symmetric Encryption. In *2023 IEEE Symposium on Security and Privacy (SP)*. 44.

[22] Josef Hadar and William R Russell. 1969. Rules for ordering uncertain prospects. *The American economic review* 59, 1 (1969), 25–34.

[23] Lingxin Hao and Daniel Q Naiman. 2007. *Quantile regression*. Sage.

[24] Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics* (1979), 65–70.

[25] Jason Hsu. 1996. *Multiple comparisons: theory and methods*. CRC Press.

[26] Mohammad Saiful Islam, Mehmet Kuzu, and Murat Kantarcioglu. 2012. Access pattern disclosure on searchable encryption: Ramification, attack and mitigation. In *Network and Distributed System Security Symposium (NDSS)*.

[27] Seny Kamara, Abdelkarim Kati, Tarik Moataz, Thomas Schneider, Amos Treiber, and Michael Yonli. 2022. SoK: Cryptanalysis of Encrypted Search with LEAKER - A framework for LEakage AttacK Evaluation on Real-world data. In *IEEE European Symposium on Security and Privacy (EuroS&P'22)*. IEEE.

[28] Seny Kamara and Tarik Moataz. 2023. Bayesian Leakage Analysis: A Framework for Analyzing Leakage in Encrypted Search. https://eprint.iacr.org/2023/813

[29] Gopal K. Kanji. 2006. *100 statistical tests* (3rd ed ed.). Sage Publications.

[30] Georgios Kellaris, George Kollios, Kobbi Nissim, and Adam O'neill. 2016. Generic attacks on secure outsourced databases. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 1329–1340.

[31] Bryan Klimt and Yiming Yang. 2004. Introducing the Enron corpus.. In *CEAS*.

[32] Roger Koenker and Gilbert Bassett. 1978. Regression quantiles. *Econometrica* (1978), 33–50.

[33] Roger Koenker and Kevin F. Hallock. 2001. Quantile Regression. *Journal of Economic Perspectives* 15, 4 (Dec. 2001), 143–156. https://doi.org/10.1257/jep.15.4.143

[34] Evgenios M. Kornaropoulos, Nathaniel Moyer, Charalampos Papamanthou, and Alexandros Psomas. 2022. Leakage Inversion: Towards Quantifying Privacy in Searchable Encryption. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22)*. Association for Computing Machinery, 1829–1842. https://doi.org/10.1145/3548606.3560593

[35] Marie-Sarah Lacharité, Brice Minaud, and Kenneth G Paterson. 2018. Improved reconstruction attacks on encrypted data using range query leakage. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 297–314.

[36] Steven Lambregts, Huanhuan Chen, Jianting Ning, and Kaitai Liang. 2022. VAL: Volume and Access Pattern Leakage-Abuse Attack with Leaked Documents. In *Computer Security – ESORICS 2022 (Lecture Notes in Computer Science)*, Vijayalakshmi Atluri, Roberto Di Pietro, Christian D. Jensen, and Weizhi Meng (Eds.). Springer International Publishing, 653–676. https://doi.org/10.1007/978-3-031-17140-6_32

[37] Chang Liu, Liehuang Zhu, Mingzhong Wang, and Yu-An Tan. 2014. Search pattern leakage in searchable encryption: Attacks and new construction. *Information Sciences* 265 (2014), 176–188.

[38] Rupert G. Jr Miller. 2012. *Simultaneous Statistical Inference*. Springer Science & Business Media.

[39] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* 267 (Feb. 2019), 1–38. https://doi.org/10.1016/j.artint.2018.07.007

[40] Jianting Ning, Xinyi Huang, Geong Sen Poh, Jiaming Yuan, Yingjiu Li, Jian Weng, and Robert H Deng. 2021. LEAP: Leakage-Abuse Attack on Efficiently Deployable, Efficiently Searchable Encryption with Partially Known Dataset. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 2307–2320.

[41] Jianting Ning, Jia Xu, Kaitai Liang, Fan Zhang, and Ee-Chien Chang. 2018. Passive attacks against searchable encryption. *IEEE Transactions on Information Forensics and Security* 14, 3 (2018), 789–802.

[42] Simon Oya and Florian Kerschbaum. 2021. Hiding the Access Pattern is Not Enough: Exploiting Search Pattern Leakage in Searchable Encryption. In *30th USENIX Security Symposium (USENIX Security 21)*.

[43] Simon Oya and Florian Kerschbaum. 2022. IHOP: Improved Statistical Query Recovery against Searchable Symmetric Encryption through Quadratic Optimization. In *31st USENIX Security Symposium (USENIX Security 22)*. 2407–2424. https://www.usenix.org/conference/usenixsecurity22/presentation/oya

[44] Rishabh Poddar, Stephanie Wang, Jianan Lu, and Raluca Ada Popa. 2020. Practical Volume-Based Attacks on Encrypted Databases. *arXiv:2008.06627 [cs]* (Aug. 2020). http://arxiv.org/abs/2008.06627

[45] Martin F Porter. 1980. An algorithm for suffix stripping. *Program* (1980).

[46] David Pouliot and Charles V Wright. 2016. The shadow nemesis: Inference attacks on efficiently deployable, efficiently searchable encryption. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 1341–1352.

[47] C Radhakrishna Rao. 1945. Information and accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society* 37 (1945), 81–91.

[48] Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. 2006. Effects of age and gender on blogging.. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, Vol. 6. 199–205.

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

[50] Shi-Feng Sun, Ron Steinfeld, Shangqi Lai, Xingliang Yuan, Amin Sakzad, Joseph Liu, Surya Nepal, and Dawu Gu. 2021. Practical Non-Interactive Searchable Encryption with Forward and Backward Privacy. In *Proceedings 2021 Network and Distributed System Security Symposium*. Internet Society. https://doi.org/10.14722/ndss.2021.24162

[51] Shi-Feng Sun, Xingliang Yuan, Joseph K. Liu, Ron Steinfeld, Amin Sakzad, Viet Vo, and Surya Nepal. 2018. Practical Backward-Secure Searchable Encryption from Symmetric Puncturable Encryption. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18)*. Association for Computing Machinery, 763–780. https://doi.org/10.1145/3243734.3243782

[52] Alexey Tsymbal. 2004. The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin* 106, 2 (2004), 58.

[53] Viet Vo, Xingliang Yuan, Shi-Feng Sun, Joseph K. Liu, Surya Nepal, and Cong Wang. 2023. ShieldDB: An Encrypted Document Database With Padding Countermeasures. *IEEE Transactions on Knowledge and Data Engineering* 35, 4 (April 2023), 4236–4252. https://doi.org/10.1109/TKDE.2021.3126607

[54] Eric W Weisstein. 2004. Bonferroni correction. *https://mathworld. wolfram. com/* (2004).

[55] Lei Xu, Huayi Duan, Anxin Zhou, Xingliang Yuan, and Cong Wang. 2021. Interpreting and Mitigating Leakage-abuse Attacks in Searchable Symmetric Encryption. *IEEE Transactions on Information Forensics and Security* (2021).

[56] Lei Xu, Xingliang Yuan, Cong Wang, Qian Wang, and Chungen Xu. 2019. Hardening database padding for searchable encryption. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2503–2511.

[57] Lei Xu, Leqian Zheng, Chengzhi Xu, Xingliang Yuan, and Cong Wang. 2023. Leakage-Abuse Attacks Against Forward and Backward Private Searchable Symmetric Encryption. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS' 23)*. Association for Computing Machinery.

[58] Xianglong Zhang, Wei Wang, Peng Xu, Laurence T. Yang, and Kaitai Liang. 2023. High Recovery with Fewer Injections: Practical Binary Volumetric Injection Attacks against Dynamic Searchable Encryption. In *32nd USENIX Security Symposium (USENIX Security 23)*.

[59] Yupeng Zhang, Jonathan Katz, and Charalampos Papamanthou. 2016. All your queries are belong to us: The power of file-injection attacks on searchable encryption. In *25th USENIX Security Symposium (USENIX Security 16)*. 707–720.

[60] Indrė Žliobaitė, Mykola Pechenizkiy, and João Gama. 2016. An Overview of Concept Drift Applications. In *Big Data Analysis: New Algorithms for a New Society*, Nathalie Japkowicz and Jerzy Stefanowski (Eds.). Vol. 16. Springer International Publishing, Cham, 91–114. https://doi.org/10.1007/978-3-319-26989-4_4

A UNIFORM DOCUMENT SET SPLITTING, A FAVORED ATTACKER SIMULATION

This appendix shows that the classic attack simulation using uniform splitting for document set generation creates the best-case scenario for the attacker. Hence, any result obtained with uniform splitting is, on average, greater than or equal to the accuracy of a real-world attacker. We prove it in three steps: *(1)* uniform splitting produces dataset distributions with equal co-probabilities, *(2)* equal co-probabilities lead to smaller ϵ-similarity, *(3)* smaller ϵ leads to higher accuracy. This appendix focuses on proving analytically the second step, and we rely on auxiliary results for steps 1 and 3.

Appendix B covers the first step using a statistical test. Indeed, the experiments show that the equality of co-probabilities is not rejected using uniform splitting. This statement could also be proven analytically using basic probability notions, but it is not the focus of this appendix.

Previous works [12, 15] experimentally proved the third step for the refined score attack. Since the ϵ measures the quality of the attacker knowledge, we can interpret the statement as follows: more precise attacker knowledge leads to higher accuracy. We can be convinced this statement is true for all attacks because the opposite sounds non-sensical.

A.1 Definitions

Notation. Let $\mathbb{E}[X]$ be the expected value of the random variable X, and $\mathrm{Var}(X)$ be its variance. The sign $\xrightarrow{\mathbb{P}}$ denotes the convergence in probability, and the sign $\xrightarrow{(d)}$ the convergence in distribution. We use the notation δ_a for the Dirac distribution on point $a \in \mathbb{R}$.

Stochastic dominance definition. Our mathematical analysis relies on "stochastic dominance" [22], a partial order between random variables. Let the sign $\preccurlyeq$ define this partial order:

Definition 2. *A random variable A has first-order stochastic dominance over random variable B if* $\forall x, \mathbb{P}(A \leq x) \geq \mathbb{P}(B \leq x)$.

Our analysis focuses only on first-order stochastic dominance. We rely on the alternative definition presented in Definition 3 to prove the stochastic dominance.

Definition 3. *A random variable A has first-order stochastic dominance over random variable B if, for a utility function u continuous, bounded, and increasing, we have* $\mathbb{E}[u(A)] \geq \mathbb{E}[u(B)]$.

ϵ-similarity distribution Let $\mathcal{E}^{p_{\text{ind}},p_{\text{atk}}}$ define the random probability distribution of the ϵ-similarity. Since $\epsilon = \|\text{SimMat}\|$, we have $\mathcal{E}^{p_{\text{ind}},p_{\text{atk}}} = \left\| \frac{C^{n_{\text{ind}},p_{\text{ind}}}}{n_{\text{ind}}} - \frac{C^{n_{\text{atk}},p_{\text{atk}}}}{n_{\text{atk}}} \right\|$.[11] The parameters p_{ind} and p_{atk} are the parameters of the probability distributions from which D_{ind} and D_{atk} are drawn.

[11] This equation manipulates random probability distributions $C^{n_{\text{atk}},p_{\text{atk}}}$ and $C^{n_{\text{ind}},p_{\text{ind}}}$ contrary to Equation (1) that manipulates classic matrices.

A.2 Stochastic advantage

We want to show that having equal co-probabilities on the attacker and the indexed document sets leads to smaller ϵ. Mathematically, we want to compare the following random distributions: $\mathcal{E}^{p_{\text{ind}},p_{\text{ind}}}$ (i.e., equality of co-probabilities) and $\mathcal{E}^{p_{\text{ind}},p_{\text{atk}}}$. Appendix A.3 proves that asymptotically (when the document set sizes tend to infinity):

$$\mathcal{E}^{p_{\text{ind}},p_{\text{ind}}} \preccurlyeq \mathcal{E}^{p_{\text{ind}},p_{\text{atk}}} \tag{5}$$

This dominance result implies that it is more likely to reach lower ϵ values when the attacker document set is drawn from a distribution with the same probabilities p_{ij} as the distribution from which the indexed document set has been drawn.

A.3 Stochastic dominance proof

The previous notations were simplified, so we redefine them more precisely here. First, the document set sizes are now considered random variables[12]. Let $\left(n_n^{\text{ind}}\right)_n$ be a sequence of random variables such that $\lim_{n\to+\infty} n_n^{\text{ind}} = +\infty$. Analogously, we have a sequence of random variables $\left(n_n^{\text{atk}}\right)_n$ be a sequence of random variables such that $\lim_{n\to+\infty} n_n^{\text{atk}} = +\infty$. We can redefine the distribution of the co-occurrence matrices from these variables: $\forall i, j \in \{1 \dots m\}$, let $C_{ij,n}^{\text{ind},p^{\text{ind}}} \sim \mathcal{B}\left(n_n^{\text{ind}}, p_{ij}^{\text{ind}}\right)$ (resp. $C_{ij,n}^{\text{atk},p^{\text{atk}}} \sim \mathcal{B}\left(n_n^{\text{atk}}, p_{ij}^{\text{atk}}\right)$) be the random probability distribution of the co-occurrence matrix of the indexed (resp. attacker) document set. We note $C_{\cdot,n}^{\text{ind},p^{\text{ind}}}$ (resp. $C_{\cdot,n}^{\text{atk},p^{\text{atk}}}$) the complete matrix distribution (i.e., $C_{ij,n}^{\text{ind},p^{\text{ind}}}$ is the distribution of the i, j variable of $C_{\cdot,n}^{\text{ind},p^{\text{ind}}}$). We assume that $C_{ij,n}^{\text{ind},p^{\text{ind}}}$ and $C_{ij,n}^{\text{atk},p^{\text{atk}}}$ are independent, but we do not suppose independence for $C_{ij,n}^{\text{ind},p^{\text{ind}}}$ and $C_{i'j',n}^{\text{ind},p^{\text{ind}}}$ (same for $C_{ij,n}^{\text{atk},p^{\text{atk}}}$ and $C_{i'j',n}^{\text{atk},p^{\text{atk}}}$). The ϵ-similarity probability distribution is then:

$$\mathcal{E}_n^{p^{\text{ind}},p^{\text{atk}}} = \left\| \frac{C_{\cdot,n}^{\text{ind},p^{\text{ind}}}}{n_n^{\text{ind}}} - \frac{C_{\cdot,n}^{\text{atk},p^{\text{atk}}}}{n_n^{\text{atk}}} \right\|_2$$

Using this notation, we can write Theorem 2 (equivalent to Equation (5)).

Theorem 2. *Asymptotically, we have* $\mathcal{E}_n^{p^{ind},p^{ind}} \preccurlyeq \mathcal{E}_n^{p^{ind},p^{atk}}$.

Proof. We have $\mathbb{E}\left[\frac{C_{ij,n}^{\text{ind},p^{\text{ind}}}}{n_n^{\text{ind}}}\right] = p_{ij}^{\text{ind}}$ and $Var\left(\frac{C_{ij,n}^{\text{ind},p^{\text{ind}}}}{n_n^{\text{ind}}}\right) \to 0$ when $n \to \infty$. Using Chebyshev's inequality, we deduce that

$$\frac{C_{ij,n}^{\text{ind},p^{\text{ind}}}}{n_n^{\text{ind}}} \xrightarrow[n\to\infty]{\mathbb{P}} p_{ij}^{\text{ind}} \tag{6}$$

Analogously, we obtain:

$$\frac{C_{ij,n}^{\text{atk},p^{\text{atk}}}}{n_n^{\text{atk}}} \xrightarrow[n\to\infty]{\mathbb{P}} p_{ij}^{\text{atk}} \tag{7}$$

Then, we use the continuous mapping theorem with the continuous function $f(x,y) = (x-y)^2$ on Equations (6) and (7) to obtain:

$$\left(\frac{C_{ij,n}^{\text{atk},p^{\text{atk}}}}{n_n^{\text{atk}}} - \frac{C_{ij,n}^{\text{ind},p^{\text{ind}}}}{n_n^{\text{ind}}}\right)^2 \xrightarrow[n\to\infty]{\mathbb{P}} \left(p_{ij}^{\text{atk}} - p_{ij}^{\text{ind}}\right)^2 \tag{8}$$

[12] This hypothesis is not restrictive: a deterministic sequence is a special case of a sequence of random variables.

We reuse the continuous mapping on this result but with the continuous function $g(z) = \sqrt{\sum_{k=1} z_k}$ to obtain:

$$\sqrt{\sum_{(i,j)\in\{1\dots m\}^2}\left(\frac{C_{ij,n}^{\text{atk},p^{\text{atk}}}}{n_n^{\text{atk}}} - \frac{C_{ij,n}^{\text{ind},p^{\text{ind}}}}{n_n^{\text{ind}}}\right)^2} \xrightarrow[n\to\infty]{\mathbb{P}} \sqrt{\sum_{(i,j)\in\{1\dots m\}^2}\left(p_{ij}^{\text{atk}} - p_{ij}^{\text{ind}}\right)^2} \tag{9}$$

$$\iff \mathcal{E}_n^{p^{\text{atk}},p^{\text{ind}}} \xrightarrow[n\to\infty]{\mathbb{P}} \left\|p^{\text{atk}} - p^{\text{ind}}\right\| \tag{10}$$

$$\implies \mathcal{E}_n^{p^{\text{atk}},p^{\text{ind}}} \xrightarrow[n\to\infty]{(d)} \delta_{\|p^{\text{atk}}-p^{\text{ind}}\|} \tag{11}$$

Let u be a utility function u continuous, bounded, and increasing. From the definition of the convergence in distribution, we deduce that:

$$\lim_{n\to\infty}\mathbb{E}\left[u\left(\mathcal{E}_n^{p^{\text{atk}},p^{\text{ind}}}\right)\right] = \int u d\delta_{\|p^{\text{atk}}-p^{\text{ind}}\|} \tag{12}$$

$$= u(\left\|p^{\text{atk}} - p^{\text{ind}}\right\|) \tag{13}$$

We remark that $\lim_{n\to\infty}\mathbb{E}\left[u\left(\mathcal{E}_n^{p^{\text{ind}},p^{\text{ind}}}\right)\right] = u(0)$ and that u is increasing so:

$$u(0) \leq u(\left\|p^{\text{atk}} - p^{\text{ind}}\right\|) \tag{14}$$

$$\iff \lim_{n\to\infty}\mathbb{E}\left[u\left(\mathcal{E}_n^{p^{\text{ind}},p^{\text{ind}}}\right)\right] \leq \lim_{n\to\infty}\mathbb{E}\left[u\left(\mathcal{E}_n^{p^{\text{atk}},p^{\text{ind}}}\right)\right] \tag{15}$$

From Equation (15), we use the Definition 3 of stochastic dominance to conclude that asymptotically: $\mathcal{E}_n^{p^{\text{ind}},p^{\text{ind}}} \preccurlyeq \mathcal{E}_n^{p^{\text{ind}},p^{\text{atk}}}$ □

B DOCUMENT SET SPLITTING AND EQUALITY OF CO-PROBABILITIES

Subsection 3.5 highlighted that uniform splitting could lead to more powerful attackers than temporal splitting. We want to deepen this result and highlight the difference between these two setups: the equality of co-probabilities. This appendix builds a statistical test to show that, contrary to the uniform split, the temporal split (on the Apache dataset) does not generate a document set distribution verifying the equality of co-probabilities. In other words, we rigorously prove the distribution shift over time in the Apache dataset.

Statistics background. Statistical tests help verify a hypothesis using observed data. When a statistical test is "rejected", the hypothesis is then false with high probability. These statistical tools are standard in medical research to prove various statements (e.g., vaccine efficacy).

Our goal is then to conceive a test for our equality assumption. When performing a statistical test, the analysis is focused on the p-value (with $p \in [0, 1]$). A p-value corresponds to the probability of obtaining our observed data under a certain *null hypothesis* (e.g., the equality in our case), referred to as H_0. The opposite hypothesis is referred to as H_1: the inequality of at least one of the co-probabilities in our case. The lower p is, the more confident we are that the hypothesis H_0 is false. If p is considered negligible, the null hypothesis is rejected. The negligibility threshold can be 0.01, 0.001, or even lower, depending on the research field.

The Z-test for the equality of two proportions [29] is the standard statistical test to check the equality of two probabilities. In our case, we need to test the equality of $\frac{m(m+1)}{2}$ pair of probabilities (i.e., p_{ij}^{ind} and p_{ij}^{atk}). To build our

Table 3. Statistical tests of the equality of co-probabilities on the Apache dataset using two splitting methods.

Year	2003	2005	2007	2009
$\widetilde{pv}$	0.0	0.0	0.0	0.0

(a) Temporal split

	Avg.	Min.	Max.
$\widetilde{pv}$	2.40	0.02	11.03

(b) Uniformly random split (100 repetitions).

test, we propose first to test all the pairs of co-probabilities individually and then combine the result of these tests into a unique p-value.

Individual Z-tests. We use the Z-test to individually test the equality of each co-probability p_{ij}. Each Z-test has as hypothesis $H_0 : p_{ij}^{\text{ind}} = p_{ij}^{\text{atk}}$ (and $H_1 : p_{ij}^{\text{ind}} \neq p_{ij}^{\text{atk}}$). For details about the Z-test, we refer to Kanji [29]. The Z-test for proportions is widely implemented in many languages, including R[13].

For clarity, we refer to the p-values as pv to avoid confusion with the co-probabilities already using the notation p_{ij}. Then, for each pair of keyword (w_i, w_j), we obtain a p-value pv_{ij} (for all $i, j \in [m]$). Note that $pv_{ij} = pv_{ji}$ so we have $\frac{m(m+1)}{2}$ unique p-values.

Combining multiple p-values. We now have $\frac{m(m+1)}{2}$ dependent p-values, and we need to combine all of them to create a test for our complete hypothesis stated in the equality of co-probabilities. In other words, we want a test verifying whether all the sub-hypotheses are true simultaneously, sub-hypotheses for which we have individual tests. This problem is known as the multiple comparisons problem [38]. It is non-trivial because these p-values are dependent. The simplest solution to this problem is the Bonferroni correction [2, 54].

The Bonferroni correction takes as an input M (possibly dependent) p-values and outputs a p-value for the combination of all hypotheses. The "corrected" p-value corresponds to the minimum p-value in the input set multiplied by M. We call this a *corrected* p-value because it is not formally a p-value (e.g., it can be above 1). However, it is common to interpret the output of a Bonferroni correction as a p-value. In our case, we can write the corrected p-value as follows:

$$\widetilde{pv} = \frac{m(m+1)}{2} \times \min_{(i,j)\in\{1\ldots m\}^2} pv_{ij} \tag{16}$$

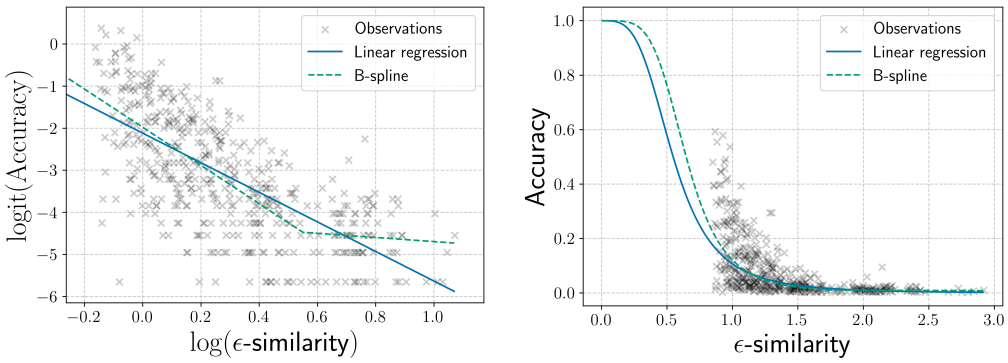
The corrected p-value is proportional to the minimum of the M initial p-values, so this test is highly sensitive. The risk of sensitive test metrics is constantly rejecting the null hypothesis for any dataset. In such a case, we could not draw any conclusions. Other solutions to the multiple comparisons problem are less sensitive than the Bonferroni correction [24, 25, 38]. However, we limited our analysis to this simple Bonferroni correction, sufficient to observe the phenomenon we want to highlight: an equality and a non-equality case.

Experimental results. To verify our distribution shift claim, we compute $\widetilde{pv}$ using the temporal split (Table 3a) and the uniform split (Table 3b) .

With the temporal split (Table 3a), the corrected p-values are extremely small, so the equality of co-probabilities is strongly rejected. Theoretically, the p-values should never be equal to zero. The values of the first experiment are so low (i.e., below machine epsilon) that they were automatically rounded to zero. This result proves the distribution shift in the Apache dataset.

Using the uniform split (Table 3b), the equality of co-probabilities is not rejected because the p-values are generally high. While we can obtain this equality result from basic mathematical analysis, this non-rejection also proves that

[13] https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/prop.test



(a) With logit – log transformation

(b) In the initial space

Fig. 9. Estimation average accuracy function on the Enron dataset with a large keyword universe ($m = 4K$)

the rejection for the temporal split is due to the splitting method properties and not the test sensitivity. We repeat the uniform split 100 times to show the temporal split was not somehow "unlucky": even the uniform split with the worst $\widetilde{pv}$ (denoted as Min. $\widetilde{pv}$ in Table 3b) has a p-value way above all the results obtained using the temporal split.

C TAIL DISTRIBUTION AND AVERAGE ACCURACY REGRESSION

During our study, we observed unexpected results when estimating the accuracy function on "extreme cases". In Figure 9, we present the regression results for the Enron dataset with a particularly large ϵ-similarity (obtained using larger keyword universes). Contrary to the rest of the experiments, the logit – log transformation does not produce a linear relationship between the dimensions. In Figure 9a, we observe for $\log(\epsilon)$ in $[-0.2, 0.5]$ the linear relationship observed in the other experiments. However, this linearity stops on the tail (i.e., $\epsilon \in [0.5, \infty)$).

This phenomenon is caused by the fact that the average accuracy cannot be below $\frac{1}{m}$, the success probability of a random guess. Hence, we should represent the average accuracy function in two pieces: one for the "small" ϵ-similarities and one for the large ϵ-similarities (i.e., the tail distribution). As the average accuracy cannot be below $\frac{1}{m}$, the average accuracy should always be $\frac{1}{m}$ for such extreme ϵ-similarities.

To deal with this phenomenon, we propose two solutions. First, we can use a regression model that is more complex than linear regression. Figure 9 compares the estimation done with the linear regression to a B-spline. B-splines are regression models that separate the space into several chunks and estimate a distinct polynomial function (in our case, linear) for each chunk. Second, we can remove the tail results from the training dataset. Indeed, the tail corresponds to negligibly small attack results. The error made by ignoring these points can seem large in the logit – log space, but in our initial space, Figure 9b shows the difference is negligible (i.e., below 1%).

D SAMPLING ALGORITHMS

Figure 10 presents the optimized document size sampling used in Section 5. This optimization aims to minimize the number of simulations while maximizing the space our simulation results cover. This optimized variant assumes that $n_{\text{atk}} = n_{\text{ind}}$ because Subsection 3.3 showed that they have a symmetrical influence on ϵ. Our algorithm introduces two new parameters influencing the coverage of the ϵ domain: the number of experiments NB_EXP and the minimum

Require: D: simulation document set with $n = |D|$; $params$: fixed attack parameters (e.g., keyword universe or query set size); NB_EXP: number of experiments; min_docs: minimum number of documents.
Ensure: $results$, a list of tuples (ϵ, Acc).
  $results \leftarrow []$
  $s_{\min} = \frac{4}{n}$
  $s_{\max} = \frac{2}{\text{min\_docs}}$
  for $k \in \{1 \dots \text{NB\_EXP}\}$ **do**
    $s_{\text{curr}} = (s_{\max} - s_{\min}) \times \frac{2i}{100} + s_{\min}$
    $n_{\text{curr}} = \left\lfloor \frac{2}{s_{\text{curr}}} \right\rfloor$
    $D_{\text{ind}} = \text{UnifSample}(D, n_{\text{curr}})$
    $D_{\text{atk}} = \text{UnifSample}(D \backslash D_{\text{ind}}, n_{\text{curr}})$
    $\epsilon = \text{Similarity}(D_{\text{atk}}, D_{\text{ind}})$
    $\text{Acc} = \text{SimulateAtk}(D_{\text{ind}}, D_{\text{atk}}, params)$
    Append (ϵ, Acc) to $results$
  end for

Fig. 10. Optimized procedure to generate results with varying ϵ-similarity

Require: D_{ex}: sample document set with $n = |D_{\text{ex}}|$; $params$: fixed index parameters (e.g. keyword universe); NB_EXP: number of experiments; A similar-data attack atk to be simulated; A quantile level α.
Ensure: Parameters of the quantile regression model (a, b)
  Minimum size of a document set: $\text{min\_size} = 0.05 * n$
  $\text{max\_sum} = 2/\text{min\_size}$
  $\text{min\_sum} = 1/n$ {Minimum value of the sum $\frac{1}{n_{\text{atk}}} + \frac{1}{n_{\text{ind}}}$}
  Initialize xy as an empty list
  for $i \in \{1 \dots \text{NB\_EXP}\}$ **do**
    $\text{curr\_sum} = (\text{max\_sum} - \text{min\_sum}) \times \frac{2i}{100} + \text{min\_sum}$
    $n_{\text{curr}} = \lfloor 2/\text{curr\_sum} \rfloor$
    $D_{\text{ind}} = \text{UnifSample}(D_{\text{ex}}, n_{\text{curr}})$
    $D_{\text{atk}} = \text{UnifSample}(D_{\text{ex}} \backslash D_{\text{ind}}, n_{\text{curr}})$
    $\text{curr\_acc} = \text{SimulateAttack}(atk, D_{\text{ind}}, D_{\text{atk}}, params)$
    Append $(\log(2/n_{\text{curr}}), \text{logit}(\text{curr\_acc}))$ to xy
  end for
  $(a, b) \leftarrow \text{QuantileRegression}(xy, \alpha)$

Fig. 11. Procedure to estimate the quantile function of an attack using Quantile Regression

number of documents min_docs. The uniform sampling of the ϵ space is visible in Figure 2b contrary to the non-uniform sampling obtained in Figure 6c (i.e., there are more results for smaller similarities). We also use this optimization in the risk assessment protocol of Figure 11.

We detail the quantile estimation procedure in Figure 11. This straightforward algorithm takes as inputs the sample document set, the index parameters, a quantile level, the number of experiments NB_EXP, and the attack. In this algorithm, a loop picks NB_EXP different document sizes and simulates the attack with randomly sampled document sets of such size. We then use the results of the attack simulations to compute the quantile regression described in the previous paragraph. Our algorithm uses the same optimized sampling strategy as Figure 10.