

USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Your Keywords Know Each Other: Breaking SSE with <1% Leaked Documents

Mingyu Bian, Jiabei Wang, Dandan Xu, Guangyu Huang, and Yongbin Zhou,
School of Cyber Science and Engineering, Nanjing University of Science and Technology

https://www.usenix.org/conference/usenixsecurity26/presentation/bian-mingyu

This paper is included in the Proceedings of the 35th USENIX Security Symposium.

August 12–14, 2026 • Baltimore, MD, USA

ISBN 978-1-939133-58-8

Open access to the Proceedings of the 35th USENIX Security Symposium is sponsored by



جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

ARTIFACT EVALUATED
USENIX
AVAILABLE

Your Keywords Know Each Other: Breaking SSE with <1% Leaked Documents

Mingyu Bian[1] Jiabei Wang[1, ✉] Dandan Xu[1]
Guangyu Huang[1] Yongbin Zhou[1, ✉]

[1]*School of Cyber Science and Engineering, Nanjing University of Science and Technology, China*
{saunafish, wangjiabei, xudandan, huangguangyu, zhouyongbin}@njust.edu.cn

Abstract

Searchable symmetric encryption (SSE) schemes inevitably leak search, access, and volume patterns, which adversaries can exploit along with partial knowledge of the data to recover documents and queries. Prior passive attacks rely on document-keyword occurrence matching, but under realistic low leakage, many keywords induce identical patterns, forming large ambiguous groups that sharply limit recovery.

We observe this failure arises from missing relational signals among keywords. Keywords, however, exhibit stable semantic regularities across corpora even when direct co-occurrence is sparse, allowing external linguistic knowledge to augment observed leakage. Based on this insight, we present Whisper, a staged co-occurrence augmented query recovery framework that weaves LLM-synthesized corpora with pre-trained embeddings to densify co-occurrence matrices and disambiguate otherwise indistinguishable candidates.

We evaluate Whisper on six diverse real-world datasets, including three collected in late 2025 to test generalization beyond LLM training cutoffs. At 0.1% leakage, previously considered safe, Whisper boosts correct query recovery from 17–27% under current SOTA to 36–49% across all datasets, a 2–2.5× improvement. Even at 1% leakage, Whisper approaches near-complete recovery, exceeding 80% on most datasets and peaking at 95%, while remaining effective against padding and volume hiding defenses, exposing a previously underestimated vulnerability in SSE.

1 Introduction

Encrypted cloud storage has become a cornerstone of modern enterprise and personal computing, yet even fully encrypted databases are vulnerable to subtle privacy leaks. Searchable Symmetric Encryption (SSE) enables efficient keyword searches over encrypted data [12,33], but this efficiency comes at a cost: query execution inevitably exposes patterns to the server. Among the most critical are the *access pattern*, revealing which documents match each query; the *search pattern*, indicating whether two queries are identical; and the *volume pattern*, disclosing result sizes. Prior research has shown that adversaries can exploit these patterns to reconstruct plaintext queries, posing concrete privacy risks in practical SSE deployments [9, 13, 18, 25–28, 31].

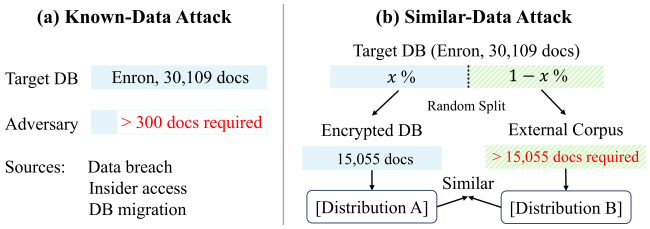


Figure 1: Illustration of typical attack assumptions.

Leakage-abuse attacks (LAA) typically rely on auxiliary knowledge to link observed patterns to plaintext keywords. Figure 1 shows that two common assumptions underlie these attacks. The known-data assumption posits that the adversary has access to a subset of the target database, acquired via data breaches, insider access, or database migration. Attacks such as IKK [18], Count [9], VAL [22], and LEAP [26] operate under this assumption. But they generally require over 1%-5% of the database, with some needing more than 90% to achieve meaningful recovery, a requirement rarely met in practice.

The similar-data assumption relaxes the requirement of accessing plaintext documents and instead assumes that the adversary can obtain a similar document set. Under this model, attacks recover queries by exploiting statistical information derived from the similar dataset, such as keyword frequency and the probability of two keywords co-occurring in the same document, as demonstrated by [25, 27, 28, 31]. However, this apparent practicality hinges on a restrictive premise: the availability of a surrogate dataset whose statistical structure closely aligns with that of the target database. In practice, this assumption is often instantiated through idealized experimental

✉ Corresponding authors.

setups. For example, Jigsaw [25] splits the original corpus in half, using one portion as auxiliary knowledge while encrypting the other as the attack target. Likewise, RSA [13] assumes that the server stores 60% of the corpus while the adversary has direct access to the remaining 40%. These setups implicitly presume statistical alignment and data availability that are difficult to justify in realistic scenarios, where auxiliary information is typically fragmented, limited, and noisy.

Motivated by this gap, we revisit the known-data threat model and consider a more realistic regime: low-leakage known-data attacks, where the adversary observes less than 1% of the target database through partial breaches, residual migration data, or limited insider access. Existing attacks fail in this regime due to a common bottleneck: co-occurrence matrix sparsity. Methods such as IKK [18] and Jigsaw [25], rely on matrices K capturing pairwise keyword frequencies. With minimal data, these matrices are mostly empty. For instance, in a 30,000-document database where only 30 documents are leaked (0.1%), observed keyword pairs cover less than 2% of all possible pairs. Most keywords never co-occur, leaving them indistinguishable and stalling attacks. Even the state-of-the-art VAL [22], which leverages document occurrence patterns and frequency counts, recovers fewer than 20% of keywords at a 0.1% leakage rate.

Our key insight is that general linguistic knowledge can systematically compensate for sparsity in low-leakage settings. Rather than depending on hard-to-obtain domain-specific corpora, readily available semantic knowledge can enrich sparse co-occurrence matrices. Building on this insight, we introduce Whisper, a novel attack framework that reconstructs discriminative keyword structures through a three-stage process. First, it identifies a small set of high-confidence keyword matches as semantic anchors. It then leverages these anchors to guide co-occurrence matrix augmentation, injecting plausible relationships via large language models (LLMs)-generated synthetic documents and embedding-based semantic similarity, yielding an enriched matrix K'_{aug}. Finally, Whisper resolves the residual ambiguity induced by noisy semantic signals through global optimization, employing the Hungarian algorithm [21] to produce a consistent keyword matching.

Contributions. We present three main contributions:

- *Designing Whisper for ultra-low-leakage attacks.* We explore a previously infeasible attack regime where the adversary observes less than 1% of the target database. This motivates Whisper, a systematic framework that reconstructs discriminative structure from sparse co-occurrence observations. Through a staged process of anchor acquisition, structure augmentation, and global matching-based disambiguation with consistency enforcement, Whisper integrates multiple weak signals to improve query recovery under ultra low-leakage conditions.
- *Incorporating semantic augmentation for sparsity compensation.* We demonstrate for the first time that modern AI tools, including LLMs and pre-trained embeddings, provide corpus-specific and general semantic signals in this extreme scarcity setting. Our fusion mechanism integrates these signals with the observed leakage to compensate for the sparsity of co-occurrence observations, improving attack performance.
- *Evaluating effectiveness across diverse datasets.* We evaluate Whisper across six real-world datasets, with three datasets collected in late 2025 to evaluate generalization beyond typical LLM training corpora. At a 0.1% leakage, Whisper achieves over 2.5× improvement in Correct Recovery Rate (CRR) compared to VAL (from 17% to 42%), while maintaining strong performance at higher leakage levels. Ablation results further indicate that even without AI-assisted signals, Whisper yields a 6%–11% improvement in CRR over the baseline. Notably, with only 9 leaked documents, Whisper can, in certain cases, achieve a CRR of up to 22.16% on Enron and 14.76% on CC-News, substantially outperforming VAL.

Table 1: Notation summary.

Category	Symbol	Description
Prior Knowledge (Plaintext)	$\mathcal{D}'$	Leaked document set $\{d_1,\ldots,d_{n'}\}$
	$\mathcal{W}''$	Leaked keywords set $\{w_1,\ldots,w_{m'}\}$
	A'	$m' \times n'$ keyword-document matrix
	M'	$n' \times n'$ document co-occurrence matrix
	K'	$m' \times m'$ keyword co-occurrence matrix
	$\text{vol}(d_i)$	Volume (bit size) of document d_i
	$\lvert d_i \rvert$	Number of keywords in document d_i
	C	Matched document pairs
Observed Leakage (Ciphertext)	$\mathcal{D}$	Plaintext document set $\{d_1,\ldots,d_n\}$
	$\mathcal{ED}$	Encrypted documents set $\{ed_1,\ldots,ed_n\}$
	$\mathcal{W}$	Keyword universe $\{w_1,\ldots,w_m\}$
	$\mathcal{Q}$	Query set $\{q_1,\ldots,q_m\}$
	B	$m \times n$ query-document matrix
	M	$n \times n$ document co-occurrence matrix
	K	$m \times m$ query co-occurrence matrix
	R	Matched query pairs

2 Background

Notation. Throughout this paper, we use d, w, ed, and q to denote plaintext documents, keywords, encrypted documents, and query tokens, respectively. Their indexed forms d_i, w_i, ed_i, and q_i refer to individual instances of each entity. Note that d (resp. w) is indexed independently from ed (resp. q). In other words, ed_i may not be the encryption of d_i, and q_i may not be the query token corresponding to w_i, even though they share the same subscript. In addition, we use (ed,d) to denote a mapping between an encrypted document and the corresponding plaintext document, and use (q,w) to denote a mapping between a query token and the corresponding keyword. The notations are given in Table 1.

Large Language Models (LLMs). Large language models are neural networks trained on massive text corpora to predict

and generate natural language [7, 34]. Given a prompt, LLMs can generate coherent, contextually relevant text that captures semantic relationships between concepts. In our attack, we leverage LLMs to synthesize documents containing target keyword combinations, thereby augmenting the attacker's auxiliary knowledge without requiring additional leaked data.

Text Embeddings. Text embedding models map words, phrases, or documents into dense vector representations in a continuous semantic space. Semantically similar texts are mapped to nearby vectors, enabling similarity computation via cosine distance. We use sentence transformer models [32] to obtain keyword embeddings, which provide external semantic knowledge about keyword relationships independent of the observed corpus. This embedding-based similarity serves as a complementary signal when co-occurrence patterns alone are insufficient for disambiguation.

3 Attack Model

We model a passive, honest-but-curious adversary whose goal is to infer sensitive plaintext from encrypted search interactions. This threat model captures the interplay between the adversary's inference goals, the leakage exposed by the protocol, its prior knowledge in the form of a small set of known data available before the attack, and its ability to leverage external tools.

Adversary Goals. The adversary aims to infer sensitive plaintext information from encrypted interactions. In particular, we focus on (i) *keyword recovery*, which targets the plaintext keywords underlying encrypted query tokens, formalized as recovering the mapping $R : q \mapsto w$, and (ii) *document recovery*, which targets the correspondence between encrypted documents and known plaintext documents, formalized as recovering $C : ed \mapsto d$.

Leakage and Observation Surface. Although the contents of queries and documents remain encrypted, the server inevitably learns certain access-related metadata, which together constitute the leakage and observation surface of the system.

Concretely, through passive observation of the encrypted document collection $\mathcal{ED} = \{ed_1, \ldots, ed_n\}$ and the query sequence $\mathcal{Q} = \{q_1, \ldots, q_m\}$, the adversary learns which encrypted documents are returned in response to each query, as well as the corresponding response volumes. Aggregating these observations enables the adversary to construct a query–document incidence matrix $B \in \{0,1\}^{m \times n}$ and to derive higher-order statistics such as query co-occurrence relationships. We formalize the resulting leakage as *Access Pattern* and *Volume Pattern*, following standard definitions [6].

Definition 1 (Access Pattern (AP))**.** *Access pattern reveals the document identifiers returned by each query. For a query q corresponding to keyword w, $AP(q) = \{id \mid w \in d_{id}\}$. This can be viewed as the rows of the query-document matrix B.*

Definition 2 (Volume Pattern (VP))**.** *Volume pattern reveals the number of documents matching each query, defined as $VP(q) = |AP(q)|$.*

Depending on the system instantiation, it may also include the bit-size of encrypted documents $\mathrm{vol}(ed_i)$. In practice, unique volume signatures often serve as anchor points for keyword inference due to Zipfian frequency distributions [22].

Prior Knowledge. We consider a known-plaintext setting in which the adversary possesses a small fraction of the plaintext document collection. Formally, the attacker is given access to a subset $\mathcal{D}' = \{d_1, \ldots, d_{n'}\} \subset \mathcal{D}$, where $|\mathcal{D}'| \ll |\mathcal{D}|$. In this paper, we focus on settings where the known-data rate (i.e., leakage rate) ranges from only 0.1% to 1% of the dataset.

This setting is more conservative than prior leakage-abuse attacks: IKK [18] requires over 90% known data for approximately 80% recovery, while VAL [22] and LEAP [26] achieve 50%–95% recovery with 1%–5% known data. For example, on the Enron dataset (30,109 documents), a 5% assumption would require access to roughly 1,500 plaintext emails, whereas a 0.1% fraction corresponds to only about 30 documents. Such documents are likely to be low-entropy and broadly distributed content (e.g., public mailing-list messages or broadcast-style notifications), which are naturally exposed to adversaries and thus constitute a realistic and accessible source of auxiliary plaintext knowledge.

From the known plaintext subset $\mathcal{D}'$, the adversary extracts a set of keywords $\mathcal{W}' = \{w_1, \ldots, w_{m'}\}$ and constructs a keyword–document incidence matrix $A' \in \{0,1\}^{m' \times n'}$, where $A'_{u,v} = 1$ indicates that keyword w_u appears in document d_v. The adversary further derives the corresponding keyword co-occurrence matrix $K' = A'A'^{\top} \in \mathbb{N}^{m' \times m'}$.

LLM Access Capabilities. We assume that the adversary has access to large language models (LLMs) as off-the-shelf generative tools. The LLM is treated strictly as a black-box generator and has no access to the target encrypted corpus, the query workload, or any plaintext documents beyond $\mathcal{D}'$. In particular, the LLM is not assumed to possess domain-specific knowledge of the target dataset. Instead, the LLM is used solely to exploit general-purpose linguistic and statistical regularities learned from public text. Conditioned only on $\mathcal{D}'$, the adversary can generate synthetic corpora that approximate keyword frequency and co-occurrence statistics.

4 Attack Design

This section presents the design of Whisper, our keyword and document recovery attack under the *low-leakage known-data* setting introduced in Section 3. We proceed in a progressive manner: we first explain why existing leakage-abuse attacks fail under our assumption, *low-leakage known data*, then present Whisper's core idea, and finally detail the complete attack algorithm.

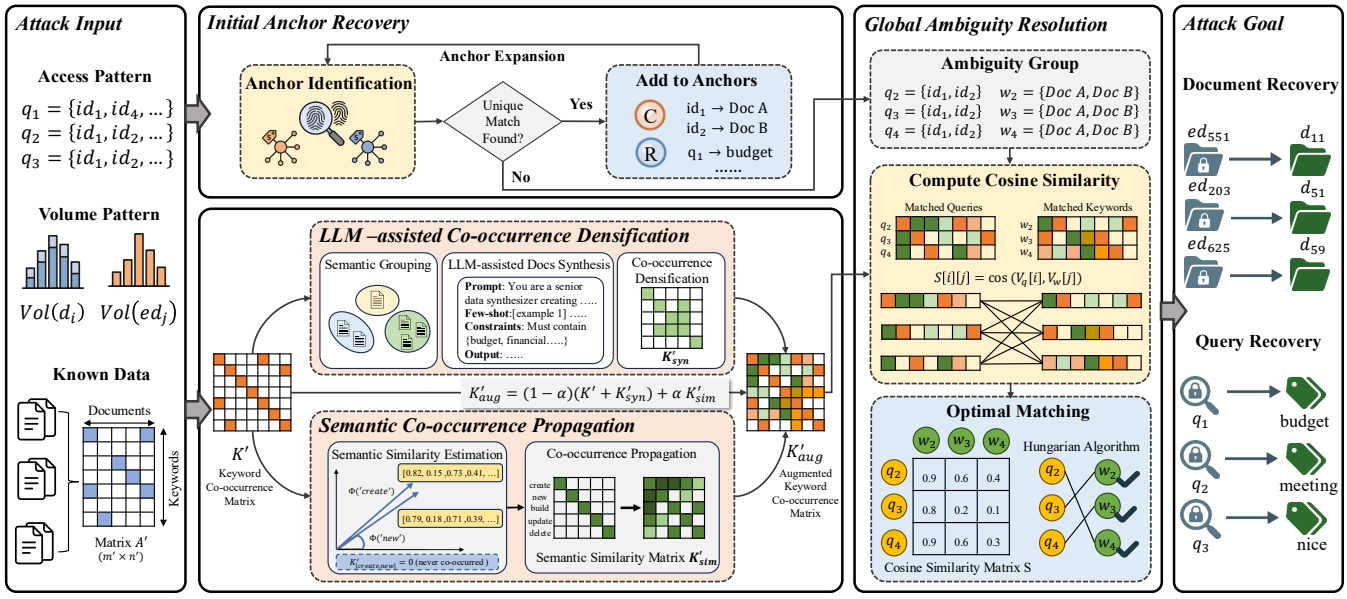


Figure 2: Attack overview of Whisper.

4.1 Attack Intuition and Overview

The central goal of known-data LAA is to recover mappings between ciphertext-side query tokens and plaintext-side keywords. Early attacks, exemplified by IKK [18], rely on the *keyword co-occurrence matrix* $K' = A'(A')^\top$, where each keyword is represented by a co-occurrence vector serving as a statistical *fingerprint* for cross-domain matching.

The effectiveness of this approach, however, hinges critically on the density of K'. Only with sufficient documents do co-occurrence vectors become discriminative. Consequently, early known-data attacks generally require non-trivial amounts of known data, with IKK as a representative case that needs over 90% to achieve recovery rates around 80%.

Subsequent work sought to relax this requirement by leveraging alternative leakage channels. Count attacks [9] exploit keyword frequency statistics, while LEAP [26] and VAL [22] incorporate access and volume patterns to extend document and keyword matching. With moderate known-data ratios of 1–5%, these approaches report recovery rates between 50% and 95%. However, once the known plaintext ratio drops below 1%, performance degrades sharply. For instance, VAL achieves only about 17% recovery at a 0.1% leakage rate. In this ultra-low-leakage regime, many keywords exhibit near-identical occurrence patterns across the limited known documents, forming large *ambiguity groups*.

Core Insight. A natural approach is to reintroduce the keyword co-occurrence matrix K' to separate ambiguous keywords. Yet under ultra-low leakage, K' remains overwhelmingly sparse and provides little discriminative power. Our key observation is that target keywords from real-world datasets are natural language tokens whose relationships follow stable semantic regularities, such as topical proximity and functional relatedness, which persist across corpora. These regularities can serve as external semantic signals to compensate for the sparsity of observed co-occurrences.

Whisper leverages this insight through two complementary mechanisms: (i) *LLM-synthesized corpora* are generated by interpolating within the co-occurrence structure implied by high-confidence keyword pairs, selectively amplifying weak but reliable empirical signals; (ii) *pre-trained word embeddings* provide soft semantic similarity priors for keyword pairs that never co-occur in the seed corpus, extending coverage beyond the empirical co-occurrence boundary.

Crucially, semantic augmentation must be carefully constrained. Unrestricted semantic signals may introduce spurious correlations that overwhelm authentic leakage-derived statistics. Whisper therefore adopts a conservative strategy: LLM synthesis is anchored on high-confidence matches and restricted to structurally plausible co-occurrences, while embedding similarities are incorporated only as soft priors within candidate ambiguity groups. Together, these mechanisms yield a densified co-occurrence matrix K'_{aug} that enhances discriminative power without significantly distorting the underlying leakage structure.

Whisper Overview. Building on these insights, Whisper first recovers high-confidence document and keyword anchors via *Initial Anchor Recovery*, exploiting leakage such as document volumes and keyword frequency patterns. For ambiguity groups unresolved by structural cues, it applies *LLM-assisted Co-occurrence Densification* and *Semantic Co-occurrence Propagation* to enrich the co-occurrence matrix, yielding K'_{aug}. Finally, globally consistent query-keyword assignments are obtained in the *Global Ambiguity Resolution* stage using co-

Algorithm 1: Whisper Attack Framework

Input: Access pattern B, Known incidence A', Volume pattern $\mathrm{vol}(\cdot)$, Known documents $\mathcal{D}'$, Blend weight α, Multiplier μ
Output: Keyword map R, document map C
$(R,C) \leftarrow \mathsf{InitialAnchorRecovery}(B,A',\mathrm{vol})$;
$K' \leftarrow A'(A')^\top$; $\quad K \leftarrow BB^\top$;
$K' \leftarrow \mathsf{LLMDensification}(\mathcal{D}',K',\mu)$;
$K'_{\mathrm{aug}} \leftarrow \mathsf{SemanticPropagation}(\mathcal{D}',K',\alpha)$;
$R \leftarrow \mathsf{GlobalAmbiguityResolution}(R,C,B,A',K'_{\mathrm{aug}},K)$;
return (R,C);

Algorithm 2: Initial Anchor Recovery

Input: Access pattern B, Known incidence A', Volume pattern $\mathrm{vol}(\cdot)$
Output: Keyword map R, document map C
$R \leftarrow \emptyset$; $C \leftarrow \emptyset$;
for $ed_j \in \mathcal{ED}, d_v \in \mathcal{D}'$ **do**
  $\sigma(ed_j) \leftarrow (\mathrm{vol}(ed_j), \|B_{:,j}\|_1)$;
  $\sigma(d_v) \leftarrow (\mathrm{vol}(d_v), \|A'_{:,v}\|_1)$;
  if $\sigma(ed_j) \approx \sigma(d_v)$ ***and*** *unique on both sides* **then**
    $C \leftarrow C \cup \{(ed_j, d_v)\}$;
$M \leftarrow B^\top B$; $M' \leftarrow (A')^\top A'$;
repeat
  for $(ed_j, d_v) \notin C$ **do**
    if $\forall (ed_k, d_u) \in C : M[j,k] = M'[v,u]$ ***and*** *unique* **then**
      $C \leftarrow C \cup \{(ed_j, d_v)\}$;
  for $q_i \in Q, w_u \in \mathcal{W}'$ **do**
    if $B[i,C] = A'[u,C]$ ***and*** *unique* **then**
      $R \leftarrow R \cup \{(q_i, w_u)\}$;
until *no change in* C *or* R;
return (R,C);

sine similarity combined with the Hungarian algorithm. Figure 2 and Algorithm 1 summarize the framework, showing how structural anchors and semantic augmentation together address low known-data leakage.

4.2 Initial Anchor Recovery

To obtain a set of highly reliable initial anchors, we adopt a widely used recovery paradigm [9, 22, 26] that iteratively extends from high-confidence matches to progressively ambiguous cases, as detailed in Algorithm 2.

Anchor Identification. Let $B \in \{0,1\}^{m \times n}$ denote the access pattern matrix, where $B_{i,j} = 1$ indicates that query token q_i matches encrypted document ed_j, and let $\mathrm{vol}(ed_j)$ denote its observable volume. From the known plaintext documents $\mathcal{D}' = \{d_1, \ldots, d_{n'}\}$, we extract the keyword set $\mathcal{W}'$ and construct the incidence matrix $A' \in \{0,1\}^{m' \times n'}$, where $A'_{u,v} = 1$ indicates that keyword w_u appears in document d_v, together with the corresponding volumes $\mathrm{vol}(d_v)$. It could be possible that not all underlying keywords are known beforehand, in which case $m' < m$, and we have to extend matrix A' to $m \times n'$ matrix that has the first m' rows equal to the original matrix A' and the following $m - m'$ rows of all 0.

We seed anchors at the document level, as document statistics are directly exposed by leakage and relatively stable. For each encrypted document ed_j and plaintext document d_v, we compute the document degree $\deg(ed_j) = \|B_{:,j}\|_1$ and $\deg(d_v) = \|A'_{:,v}\|_1$, and combine it with volume to form a compact signature $\sigma(d) = (\mathrm{vol}(d), \deg(d))$. Document pairs with signatures that are unique on both sides are deterministically matched and added to the document map C.

Starting from the recovered document correspondences, we expand C using strict consistency constraints derived from document co-occurrence. We compute the co-occurrence matrices $M = B^\top B$ and $M' = (A')^\top A'$, where each entry counts the number of queries matching a document pair. A candidate match is accepted only if its co-occurrence counts with all previously matched documents agree exactly, and is committed only when this filtering yields a unique feasible candidate.

Anchor Expansion. Given a growing set of document correspondences C, keyword recovery reduces to constrained pattern matching over the aligned document subset. For each query token and plaintext keyword, we compare their restricted occurrence vectors on the matched documents and accept a match only when the pattern is unique on both sides. Under low leakage, many keywords exhibit identical restricted patterns and form ambiguity groups. In these cases, Whisper conservatively defers resolution to the semantic refinement stage rather than committing to weak evidence.

Recovered keywords introduce new anchors that further expand the document map. We compare anchor-induced column patterns defined over the recovered tokens and add new document correspondences when a unique match exists. Document and keyword recovery are iterated in this manner until convergence, yielding a dependable initial anchor set (R,C).

4.3 LLM-assisted Co-occurrence Densification

Due to the thematic heterogeneity of the leaked plaintext corpus, naively using the entire corpus as a joint generation context causes LLMs to blend disparate discourse patterns, resulting in incoherent texts and artificial keyword co-occurrences. To address this issue, we first partition the leaked plaintext corpus into semantically coherent groups, and then extract group-level statistical characteristics to constrain LLM-assisted synthesis in a controlled manner, thereby selectively densifying the keyword co-occurrence matrix K' (see Algorithm 3).

Semantic Grouping. To decompose the leaked plaintext corpus into semantically coherent groups, we embed each document into a semantic vector space and organize the corpus using hierarchical clustering. Rather than fixing the number of groups a priori, the grouping granularity is determined adaptively based on the empirical distribution of pairwise embedding distances, yielding fewer, denser groups for topically concentrated corpora and more localized groups for heterogeneous datasets.

LLM-assisted Document Synthesis. Within each semantic group, Whisper leverages LLMs to synthesize auxiliary documents under group-specific statistical constraints. The

Algorithm 3: LLM-assisted Co-occurrence Densification

Input: Known documents $\mathcal{D}'$, Keyword co-occurrence matrix K', multiplier μ
Output: Densified matrix K'
$\mathcal{G} \leftarrow \mathsf{Cluster}(\{\mathsf{Embed}(d) : d \in \mathcal{D}'\})$;
for $g_i \in \mathcal{G}$ **do**
  $\mathcal{E} \leftarrow \mathsf{SelectExemplars}(g_i)$;
  $\mathcal{W}_k \leftarrow \mathsf{SelectTopKeywords}(g_i)$;
  $\mathcal{P}_k \leftarrow \mathsf{SelectTopPairs}(g_i)$;
  $n_t \leftarrow \mu \times |g_i|$;
  $\ell_t \leftarrow \mathsf{SampleLength}(g_i)$;
  $\mathcal{S} \leftarrow \mathsf{LLM}(\mathcal{E}, \mathcal{W}_k, \mathcal{P}_k, n_t, \ell_t)$;
  for $s \in \mathcal{S}$ **do**
    $T \leftarrow \mathsf{ExtractKeywords}(s) \cap \mathcal{W}'$;
    for $(w_i, w_j) \in T \times T$ **do**
      $K'[i,j] \leftarrow K'[i,j] + 1$;
return K';

Prompt Template

System: You are a senior data synthesizer creating realistic `[domain]` documents in `[language]`. Mimic the discourse patterns of the provided examples without copying phrases.

Reference Examples:
[Put representative seed documents here]

Constraints:
- Produce `[`n_t`]` diverse documents (∼`[`ℓ_t`]` words each)
- Must contain the following keywords:
  [Put required keywords here]
- Must include the following co-occurrence patterns:
  [Put required co-occurrence pairs here]

Output: Document bodies separated by `###`

Figure 3: Simplified prompt template for document synthesis.

purpose of this step is not to introduce new semantic evidence, but to interpolate within the co-occurrence structure implied by the limited seed documents belonging to the same group. Figure 3 presents a simplified illustration of the adaptive prompt used by Whisper.

The synthesis task and target corpus characteristics are explicitly specified in the system prompt. In particular, the `[domain]` and `[language]` parameters are manually configured according to the target dataset and define the stylistic and linguistic scope of generation. For example, when processing the Enron dataset, these parameters are instantiated as *corporate/operational email bodies* and *English*, respectively. Together with the instruction to mimic discourse patterns without copying phrases, these settings guide the LLM to emulate corpus-specific writing styles while avoiding verbatim reuse of seed content, thereby preserving both thematic consistency and lexical diversity.

To anchor generation within the semantic scope of each group, Whisper adopts a few-shot prompting strategy by automatically selecting a small set of representative documents from the corresponding semantic group g_i and providing them as reference examples in the prompt. Specifically, Whisper selects one document closest to the cluster centroid c_i of group g_i in the embedding space to capture the prototypical discourse patterns and dominant stylistic features of the group, and selects several documents farthest from c_i to capture peripheral expressions and edge-case variations. Considering the context length limitations of LLMs, Whisper includes at most ten reference documents. This selection strategy allows the LLM to observe both the semantic core of each group and its inherent stylistic variability, ensuring that generation remains anchored while avoiding overly homogeneous outputs.

In addition, Whisper models the empirical distribution of document lengths within each semantic group to align the length profiles of generated documents with those of the original corpus. For each group, a target document length ℓ_t is sampled from the group-specific distribution and injected into the prompt, guiding the LLM to generate n_t documents with comparable lengths, thereby mitigating volume-induced bias in statistics. The number of generated documents n_t is controlled by a synthesis multiplier μ such that, for a semantic group g_i, Whisper generates approximately $n_t = \mu \cdot |g_i|$ documents, scaling the synthetic data proportionally to the group size. In practice, we find that setting $\mu = 1$ is sufficient, as larger values introduce additional overhead with only marginal gains and a higher likelihood of spurious noise.

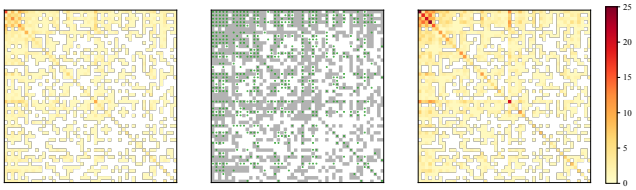
Finally, to reinforce keyword co-occurrence signals without disrupting existing relationships, Whisper further imposes explicit statistical constraints in the prompt, derived from both the topical composition of each semantic group and the observed relationships between keywords. Specifically, Whisper extracts the top-k high-frequency keywords from each group and requires their presence in the generated documents, thereby anchoring generation to the group-specific topic distribution. At the same time, high-frequency keyword co-occurrence pairs identified from the group-level co-occurrence matrix are enforced to occur jointly within each generated document. Under these constraints, the LLM is guided to output only raw document bodies, with individual documents separated by a fixed delimiter (`###`).

Co-occurrence Densification. After document synthesis, Whisper adopts a conservative integration strategy to densify the keyword co-occurrence matrix K'. Crucially, only keywords that already appear in the leaked plaintext corpus $\mathcal{W}'$ are considered; all out-of-vocabulary tokens generated by the LLM are strictly discarded. This design ensures that synthetic data can only reinforce or reveal latent relationships among known keywords, without introducing spurious vocabulary that could pollute the attack's semantic foundation. For any pair of keywords that co-occur within a generated document, the corresponding co-occurrence increment is added to the original matrix K', yielding a denser representation while preserving the integrity of the original keyword space.

Figure 4 illustrates the effect of the co-occurrence densifi-

cation process on a top-50 keyword submatrix. The left panel shows the original co-occurrence structure, the middle panel highlights newly established co-occurrence pairs introduced by LLM-assisted synthesis, and the right panel shows the resulting densified matrix. Notably, this top-50 submatrix is already relatively dense compared to the long tail of keywords. Even in this regime, the number of non-zero co-occurrence pairs increases from 1,242 (density 49.7%) to 1,730 (density 69.2%), corresponding to 488 newly introduced pairs.

Globally, the original K' contains 39,925 non-zero co-occurrence pairs with a density of 10.8%. After integration, 12,663 new pairs are introduced, increasing the total to 48,483 non-zero entries and raising the overall density to 13.2%.



(a) K' Initial (b) Enhancement (c) K' + LLM Synthesis

Figure 4: Visualization of co-occurrence densification via LLM-assisted synthesis on a top-50 keyword submatrix.

4.4 Semantic Co-occurrence Propagation

Although LLM-assisted synthesis substantially densifies the empirical keyword co-occurrence matrix, the resulting statistics remain uneven across the keyword space. Co-occurrence evidence tends to concentrate on frequently observed terms, while semantically related but less frequent keywords may remain weakly connected. Such imbalance can reduce the robustness of subsequent matching, especially when multiple candidate keywords share similar but sparsely supported co-occurrence profiles. To mitigate this issue, Whisper introduces semantic co-occurrence propagation. The key idea is to allow semantically related keywords to partially share their existing co-occurrence evidence, thereby smoothing the co-occurrence structure without introducing new keywords or fabricating unsupported relationships.

Semantic Similarity Estimation. Whisper first quantifies semantic proximity among known keywords. Each keyword $w \in \mathcal{W}'$ is mapped into a semantic vector space via an embedding function $\Phi(w)$, instantiated using SBERT [32]. In this space, keywords with similar meanings tend to be geometrically close even if they rarely co-occur in the leaked corpus. For example, keywords such as *create* and *new* often appear in similar contexts and therefore exhibit high semantic similarity despite limited direct co-occurrence.

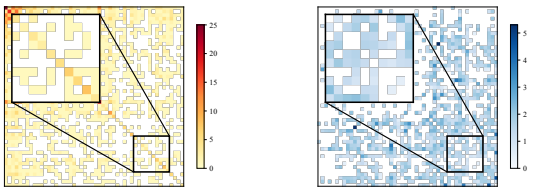
Based on these embeddings, Whisper constructs a semantic similarity matrix K_{sim} using cosine similarity:

$$K_{\text{sim}}[i,j] = \max\left(0, \frac{\Phi(w_i)\cdot\Phi(w_j)}{\|\Phi(w_i)\|\|\Phi(w_j)\|}\right). \quad (1)$$

Similarity-driven Co-occurrence Propagation. Semantic similarity is then used to guide the propagation of co-occurrence evidence. Intuitively, if a keyword w_i exhibits strong co-occurrence patterns with a set of contextual keywords, a semantically similar keyword w_j is allowed to inherit a fraction of this co-occurrence profile in proportion to their semantic proximity. This mechanism smooths sparse regions of the co-occurrence matrix without asserting that w_i and w_j must directly co-occur. Formally, Whisper constructs an augmented co-occurrence matrix as

$$K'_{\text{aug}} = (1-\alpha)K' + \alpha \cdot \text{scale}(K_{\text{sim}}), \quad (2)$$

where $\alpha \in [0,1]$ controls the strength of semantic propagation and $\text{scale}(\cdot)$ rescales semantic similarity values to be numerically comparable with empirical co-occurrence counts. We set $\alpha = 0.3$ in all experiments based on the empirical evaluation in Section 5. This formulation treats semantic similarity as a soft prior that complements, rather than overrides, leakage-derived statistics. A detailed empirical comparison of alternative fusion designs is provided in Sec. 6.3 (Table 6).



(a) K' Before Propagation (b) New Semantic Links

Figure 5: Visualization of semantic co-occurrence propagation on a top-50 keyword submatrix.

Figure 5 illustrates the effect of similarity-driven co-occurrence propagation. The left panel shows a top-50 keyword submatrix of the co-occurrence matrix before semantic propagation, where localized sparsity persists despite prior densification. The highlighted region emphasizes this phenomenon. During semantic propagation, a total of 1,052 previously weakly supported entries across the matrix receive non-zero responses guided by semantic similarity, as shown in the right panel. This propagation transforms sparse areas into more semantically coherent patterns.

4.5 Global Ambiguity Resolution

Building on the reliable initial anchors from the first phase and the co-occurrence matrix densified with LLM-generated data and external embeddings, we resolve the *ambiguity groups* as detailed in Algorithm 4.

Algorithm 4: Global Ambiguity Resolution

Input: Matched queries R, Matched docs C, Access pattern B, Known incidence A', Query co-occurrence matrix K, Augmented keyword co-occurrence matrix K'_{aug}
Output: Updated R
$\mathcal{A}_q \leftarrow \{q : (q,w) \in R\}$; $\mathcal{A}_w \leftarrow \{w : (q,w) \in R\}$;
$B[\cdot,C] \leftarrow C$ columns of B;
for $row_i \in B[\cdot,C]$ **do**
  $Q_p \leftarrow \{q \in Q \setminus \mathcal{A}_q : B[q,C] = row_i\}$;
  $\mathcal{W}_p \leftarrow \{w \in \mathcal{W}' \setminus \mathcal{A}_w : A'[w,C] = row_i\}$;
  for $q_i \in Q_p$ **do**
    $\mathbf{v}_i \leftarrow K[i, \mathcal{A}_q]$;
  for $w_j \in \mathcal{W}_p$ **do**
    $\mathbf{u}_j \leftarrow K'_{\text{aug}}[j, \mathcal{A}_w]$;
  for $q_i \in Q_p, w_j \in \mathcal{W}_p$ **do**
    $S_{i,j} \leftarrow \cos(\mathbf{v}_i, \mathbf{u}_j)$;
  $\pi^* \leftarrow \arg\max_\pi \sum_i S_{i,\pi(i)}$;
  $R \leftarrow R \cup \{(q_i, w_{\pi^*(i)}) : q_i \in Q_p\}$;
return R;

Using the recovered anchors as a semantic reference frame, we construct context vectors for each query token and corresponding vectors for candidate plaintext keywords from the augmented matrix K'_{aug}. To assign query tokens to candidate keywords, we then compute cosine similarity between each query token and candidate keyword to form a scoring matrix $S \in \mathbb{R}^{|Q|\times|W|}$ where $S_{i,j}$ quantifies semantic affinity.

A naive greedy strategy, which iteratively selects the highest-scoring pair, can fail due to competitive coupling. Consider a simple scenario with two query tokens $\{q_1, q_2\}$ and two candidate keywords $\{w_1, w_2\}$, where the similarity scores are:

$$S = \begin{pmatrix} 0.9 & 0.8 \\ 0.85 & 0.3 \end{pmatrix} \tag{3}$$

A greedy selection matches $q_1 \to w_1$ with score 0.9 and $q_2 \to w_2$ with score 0.3, yielding a total of 1.2. The globally optimal assignment $q_1 \to w_2$ and $q_2 \to w_1$ achieves 1.65, showing how greedy decisions can cascade into suboptimal outcomes.

To avoid such pitfalls, we formulate the problem as a *maximum weighted bipartite matching* task on a complete bipartite graph $G = (Q \cup W, E)$ with edge weights $S_{i,j}$, seeking a one-to-one mapping $\pi : Q \to W$ that maximizes the total weight:

$$\pi^* = \arg\max_\pi \sum_{q_i \in Q} S_{i,\pi(q_i)}. \tag{4}$$

This problem is efficiently solved using the Hungarian algorithm [21], which guarantees global optimality and prevents the error propagation inherent in greedy strategies.

5 Implementation

Datasets. We evaluate Whisper on six real-world datasets spanning multiple application domains and time periods. Detailed information about the datasets is provided in Table 2. In particular, to enable fair comparison with prior leakage abuse attacks, we include three widely used benchmark datasets: the Enron Email Corpus, consisting of 30,109 emails [10], which reflects dense corporate communication; the Apache Lucene mailing list, consisting of 27,889 messages collected between 2002 and 2025 [4], representing long-running technical discussions; and a subset of 30,000 Wikipedia articles providing general encyclopedic content [3].

To mitigate potential memorization bias introduced by LLM pre-training, we further incorporate three recent datasets collected after the training cutoff of current models. These include the Linux Kernel Mailing List (LKML), comprising 31,976 emails from October 2025 to January 2026 [1]; the QEMU virtualization project mailing list, containing 26,621 messages from September 2025 to January 2026 [2]; and a snapshot of 30,001 news articles from the Common Crawl News corpus (CC-News, January 2026) [11].

Table 2: Evaluation datasets.

Dataset	Domain	Size	Time Period
Enron	Corporate Email	30,109	1999-01–2002-12
Lucene	Tech Mailing List	27,889	2002-01–2025-12
Wikipedia	Encyclopedia	30,000	/
LKML	Kernel Dev	31,976	2025-10–2026-01
QEMU	Virtualization	26,621	2025-09–2026-01
CC-News	News Articles	30,001	2026-01–2026-02

Preprocessing. For the above datasets, we apply a unified preprocessing pipeline to obtain a consistent keyword representation. Specifically, we retain only the plain-text body content of each document or email message and discard all header metadata. We then tokenize the text, remove English stopwords based on the NLTK corpus [5], and apply Porter stemming [30] to normalize the remaining terms. Following prior leakage abuse attacks [22, 26], we rank terms by document frequency, counting each term at most once per document, and select the top 5,000 most frequent terms to construct the global keyword vocabulary $\mathcal{W}$ (see Appendix A for results with $|\mathcal{W}| \in \{1000, 3000, 5000\}$).

In addition to the above generic preprocessing, we apply an extra sanitization stage *only for the LLM-assisted synthesis module*. This stage aims to extract clean semantic content from seed documents while removing artifacts that may interfere with generation quality or trigger safety filters. For email-based datasets, we retain only the `text/plain` body content, discarding headers, quoted replies, forwarded sections, and recurring boilerplate such as unsubscribe footers. We further remove non-semantic or sensitive artifacts, including email addresses, phone numbers, URLs, file paths, timestamps, organizational directory strings, and common sensitivity markers. Finally, text is normalized by cleaning punctuation and whitespace; overly long bodies are truncated to 150 words, and very short fragments are discarded.

Prototype Implementation. We implement Whisper as an end-to-end attack prototype in Python. For synthetic docu-

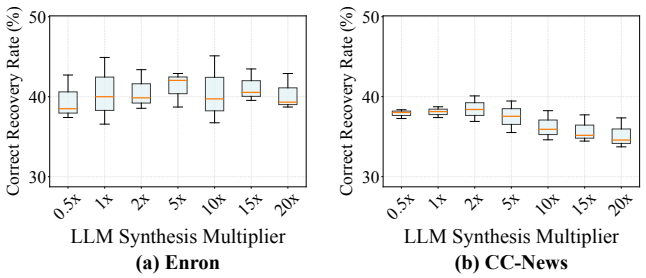


Figure 6: Effect of synthesis multiplier on CRR of query.

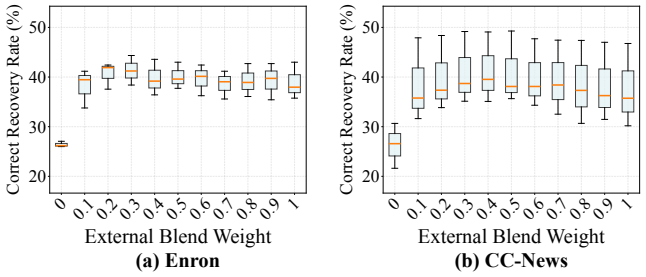


Figure 7: Effect of blend weight α on CRR of query.

ment generation, we adopt Google Gemini 2.5 Flash, selected for its strong generation quality, long context window, and a January 2025 knowledge cutoff that predates all post-cutoff evaluation datasets. The sampling temperature is set to 0.9 to encourage output diversity. During synthesis, we fix the synthesis multiplier to $\mu = 1$, generating a number of synthetic documents proportional to the size of each semantic group. This choice is empirically motivated. As shown in Figure 6, increasing μ leads to limited or unstable improvements in correct recovery rate of query, while incurring substantially higher computational and token overhead. In some cases, excessive synthesis even degrades performance due to amplified noise. Accordingly, $\mu = 1$ provides a practical balance between recovery performance and efficiency.

For semantic similarity estimation, Whisper employs the all-mpnet-base-v2 embedding model. As described in Section 4.4, the densified co-occurrence matrix is combined with the semantic similarity matrix via weighted aggregation controlled by the blend weight α. We fix $\alpha = 0.3$ based on empirical evaluation. Figure 7 shows that smaller values underutilize semantic propagation, whereas larger values overemphasize generic similarity and dilute corpus-specific structure. The selected value achieves a stable trade-off across datasets.

6 Evaluation

To evaluate the effectiveness of Whisper, we investigate the following research questions (RQs):

- **RQ1:** How effective is Whisper in recovering plaintext documents and queries across different datasets when only limited prior knowledge is available? (§6.1)
- **RQ2:** How does Whisper outperform other leakage abuse attacks? (§6.2)
- **RQ3:** Can each component contribute to Whisper? (§6.3)
- **RQ4:** How robust is Whisper against common countermeasures adopted by encrypted search systems? (§6.4)

Among these RQs, RQ1 focuses on demonstrating the effectiveness of Whisper under data-sparse attack settings. RQ2 compares Whisper with state-of-the-art approaches and investigates whether Whisper is superior to them. RQ3 conducts a comprehensive ablation study to analyze the contribution of individual components of Whisper. RQ4 studies the robustness of Whisper in the presence of commonly adopted defensive mechanisms. All experiments are conducted on a MacBook Air equipped with an Apple M4 chip, 24GB RAM, and a 512GB SSD, and each experiment is repeated three times with independently sampled runs.

Following prior leakage abuse attacks [22, 26], we evaluate attack effectiveness using three standard metrics: *Recovery Rate*, *Recovery Accuracy*, and *Correct Recovery Rate* (CRR). *Recovery Rate* measures the fraction of target items (keywords or documents) for which a mapping to query tokens or encrypted documents is successfully inferred. *Recovery Accuracy* quantifies the proportion of correct mappings among all recovered candidates. *Correct Recovery Rate* (CRR) combines these two aspects and is defined as $CRR = Recovery\ Rate \times Recovery\ Accuracy$.

6.1 Overall Attack Effectiveness

We evaluate Whisper on six real-world datasets (as detailed in Table 2) spanning diverse semantic domains to assess its ability to recover both queries and files under *ultra-low known-data* settings. Since our goal is to correctly recover as many plaintext entities as possible, Whisper always produces a complete one-to-one mapping between ciphertexts and plaintext candidates, making the overall recovery rate trivially 100%. We therefore report CRR, which measures the fraction of correctly recovered items as the known data rate varies from 0.1% to 1.0%.

As shown in Figure 8, the CRR of file remains high across all datasets, exceeding 70% even at the minimal leakage of 0.1%. In contrast, the CRR of query increases with the known data rate, confirming the strong correlation between prior knowledge and attack effectiveness. At 0.1% leakage, CRR ranges from 35.86% (QEMU) to 49.01% (Lucene), and rises to 68.74%–95.12% at 1.0%. Notably, the steepest gains occur in the ultra-low leakage regime (0.1%–0.3%), indicating that even a small number of additional known documents significantly improves the semantic co-occurrence module's ability to resolve large ambiguity groups.

The CRR of query also exhibits distinct performance tiers across datasets. General Knowledge domains (Wiki, CC-News) reach the highest recovery ceilings, around 95% at

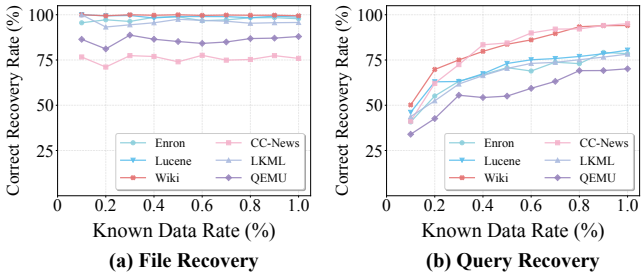


Figure 8: Overall attack results of Whisper.

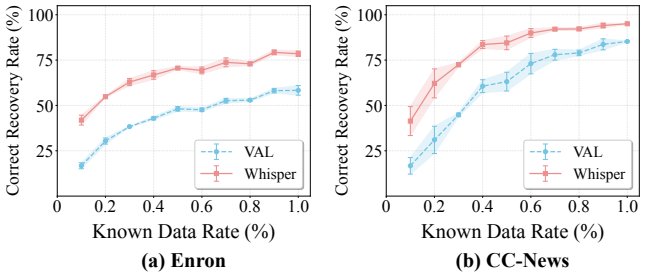


Figure 9: CRR of query: Whisper vs. VAL.

1.0% leakage. This strong performance arises from two factors: their vocabularies are well represented in general-purpose LLMs and embedding models, enabling more accurate document synthesis and semantic similarity estimation, and their keyword co-occurrence patterns follow broadly shared semantic regularities that external sources capture effectively. Technical communication datasets (Lucene, Enron, LKML) stabilize at around 80% (80.15%, 78.49%, and 78.25%), combining natural language discussions with domain-specific terminology. General semantic knowledge aids discussion-oriented content, but domain-specific jargon and project-specific acronyms introduce partial vocabulary gaps. Nonetheless, structural regularities allow Whisper's augmentation mechanisms to achieve substantial disambiguation gains. Finally, the QEMU dataset is the most challenging, achieving at most 68.74% recovery. Unlike other mailing lists covering diverse subsystems, QEMU emails are mostly code patches with highly repetitive content: nearly every message contains C constructs (`void`, `static`, `struct`, `return`) and overlapping technical terms (`device`, `memory`, `register`, `migration`). This concentrated vocabulary yields many keywords with nearly identical co-occurrence patterns, forming large ambiguity groups that challenge even semantic disambiguation. Additionally, context-specific virtualization terminology is poorly captured by general-purpose embeddings, limiting external knowledge augmentation.

These observations highlight that Whisper generalizes robustly across diverse datasets, while its effectiveness is constrained by the semantic coverage of external sources.

6.2 Comparison with State-of-the-Art Attacks

We compare Whisper with the state-of-the-art leakage abuse attack VAL [22] under varying leakage rates, using *Correct Recovery Rate* (CRR) of query recovery as the evaluation metric, which jointly reflects recovery coverage and correctness. Beyond measuring absolute performance gains, we further examine whether these improvements can be attributed to naive LLM-based augmentation of existing attacks.

Comparison with Existing Attacks. Figure 9 provides a visual comparison on Enron and CC-News, selected to illustrate the two dataset categories considered in our evaluation: widely used benchmark corpora adopted by prior work and newly collected datasets designed to mitigate potential pre-training effects. Comprehensive results across all datasets are reported in Table 3. Additional comparisons with other attacks are reported in Appendix B.

As shown in Figure 9, Whisper achieves higher CRR than VAL across all evaluated leakage rates from 0.1% to 1.0% on both datasets. Although increased leakage improves recovery performance for both methods, the performance gap between Whisper and VAL remains substantial throughout the entire range. The improvement is most pronounced in the low-leakage regime, which constitutes the most challenging setting for leakage abuse attacks. At 0.1% leakage, Whisper improves CRR by approximately 2.5× over VAL, increasing recovery from 16.74% to 41.90% on Enron and from 16.72% to 41.38% on CC-News. Comparable advantages persist at 0.2% and 0.3% leakage, indicating that co-occurrence densification is particularly effective when empirical co-occurrence evidence is sparse.

Table 3 reports detailed CRR of query recovery results of Whisper and VAL across all six datasets under varying leakage levels. Consistent with the trends observed in Figure 9, Whisper outperforms the baseline on every dataset and at every evaluated leakage rate. Notably, on the Wikipedia dataset at 0.3% leakage, VAL attains a CRR of 43.80%, whereas Whisper achieves 75.17%, corresponding to an absolute difference of 31.37%. On CC-News at 1.0% leakage, despite the narrower margin, VAL reaches 85.28%, while Whisper attains 95.03%, yielding a difference of 9.75%. Nevertheless, when averaged across leakage levels, the CRR difference remains remarkably stable, ranging from 18.04% to 24.97% across all datasets, confirming the effectiveness of Whisper across diverse corpus types.

To further evaluate robustness under *extreme* low-leakage conditions, we conduct an additional stress test where the attacker observes fewer than 0.1% of the plaintext corpus. Figure 10 reports CRR under varying numbers of leaked documents, ranging from 3 to 30. As can be seen, even with only a handful of leaked documents, Whisper achieves non-trivial CRR and consistently outperforms VAL on both Enron and CC-News. For example, with only 9 leaked documents, Whisper attains a CRR of 12.09% on Enron and 8.28% on

Table 3: CRR of query: Whisper vs. VAL across six datasets.

Leakage	Enron		CC-News		Wiki		Lucene		LKML		QEMU	
	VAL	Whisper	VAL	Whisper	VAL	Whisper	VAL	Whisper	VAL	Whisper	VAL	Whisper
0.1%	16.74	41.90	16.72	41.38	16.70	47.05	26.91	49.01	24.09	43.33	19.19	35.86
0.2%	30.37	54.81	31.00	62.16	31.18	62.34	39.21	59.40	36.55	55.80	29.86	48.08
0.3%	38.32	62.89	44.80	72.49	43.80	75.17	45.34	65.09	40.11	59.35	36.14	53.51
0.5%	48.13	70.60	63.12	84.49	58.26	84.73	51.60	71.63	49.21	69.01	41.23	59.29
0.7%	52.62	73.77	77.94	92.08	67.66	89.93	56.32	76.00	54.81	74.31	48.65	66.70
1.0%	58.31	78.49	85.28	95.03	79.16	95.12	61.38	80.15	59.69	78.25	50.89	68.74
Avg. Δ	+22.47 ↑		+19.22 ↑		+24.97 ↑		+19.89 ↑		+19.59 ↑		+18.04 ↑	

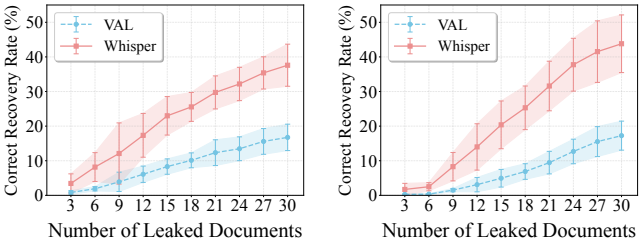


Figure 10: CRR of query under <0.1% leakage.

Table 4: Effect of naive LLM document injection.

Method	Enron (%)			CC-News (%)		
	0.1%	0.3%	0.5%	0.1%	0.3%	0.5%
VAL	15.54	37.54	47.09	16.61	44.03	63.56
VAL+LLMdocs	11.49	32.06	44.53	12.18	26.22	40.31
LEAP	14.73	35.55	45.04	16.32	43.44	63.00
LEAP+LLMdocs	10.22	35.55	45.04	16.32	43.44	38.16

CC-News, exceeding VAL by 8.19% and 6.78%, respectively. However, it can be observed that under such extreme leakage, recovery performance exhibits higher variance, as reflected by the wider confidence intervals in Figure 10 compared to Figure 9, which is expected given the limited empirical evidence available. Notably, in some runs, the leaked documents induce distinctive and stable co-occurrence profiles; for example, the numbers of keywords contained in each document are all distinct. Under this condition, even with only 9 leaked documents, Whisper can achieve a CRR of 22.16% on Enron and 14.76% on CC-News, whereas VAL reaches at most 7.06% and 2.22%, respectively. Overall, this extreme stress test shows that query recovery remains possible under severely limited leakage, albeit with increased variability.

LLM Augmentation on Prior Attacks. To evaluate whether LLM-generated data can improve existing attacks without modifying their designs, we use Whisper's synthesis pipeline to generate documents and inject them into the auxiliary corpus of VAL [22] and LEAP [26]. We report query CRR (%) on Enron and CC-News under 0.1%, 0.3%, and 0.5% leakage settings, averaged over three runs.

Table 4 reports CRR under the original setting and its +LLMdocs variant. Overall, naive LLM injection does not improve either attack. VAL consistently degrades across all settings, with losses ranging from 2.56 to 23.25 percentage points (pp). LEAP shows setting-dependent behavior: it remains unchanged in some cases where synthetic documents are filtered out, but degrades when they are retained, particularly under higher leakage.

These effects arise because synthetic documents introducing co-occurrences that are absent from the target corpus into the structural fingerprints used by both VAL and LEAP. When a synthetic plaintext is incorrectly matched as an anchor, its spurious keyword set propagates through subsequent iterations, amplifying the error. We observe up to 10 encrypted documents per run being matched to synthetic plaintexts on CC-News at 0.5% leakage. This affects both anchor identification and downstream matching, which prior attacks assume to be reliable, making additive integration ineffective.

To better utilize LLM-derived signals, prior pipelines require noise-aware integration mechanisms. Possible directions include: (i) *Confidence-based filtering*, which retains only LLM outputs supported by external evidence (e.g., embedding similarity thresholds or multi-sample agreement); (ii) *Weighted co-occurrence integration*, which combines empirical and synthetic matrices with separate weights, though performance is sensitive to parameter tuning; and (iii) *Separate-matrix fusion*, which treats LLM signals as an auxiliary source used only under ambiguity.

Under ultra-low leakage, the empirical co-occurrence matrix is often too sparse to reliably calibrate noisy signals, making the integration point critical. Prior attacks typically incorporate auxiliary data at the input stage, whereas Whisper fuses semantic and structural signals directly at the matching stage via a soft-prior formulation.

6.3 Ablation Study

We perform an ablation study to quantify how key design choices in Whisper contribute to its effectiveness, focusing on component-level contributions, semantic-fusion strategies, and global ambiguity resolution under ultra-low leakage.

Table 5: Whisper component ablation at 0.1% leakage (CRR).

Configuration	Enron (%)	CC-News (%)
VAL (baseline)	16.74	18.13
Whisper w/o AI-assisted	24.60	28.62
Whisper w/o Co-Prop.	27.61	32.87
Whisper w/o LLMSyn.	39.11	44.33
Full Whisper	40.96	46.37

Component Ablation. We evaluate five configurations under 0.1% leakage ($\approx$ 30 known emails/articles), with fixed hyperparameters ($\mu = 1.0, \alpha = 0.3$). The configurations include: the VAL baseline, Whisper without AI-assisted signals (i.e., retaining our attack structure without Co-Prop. and LLMSyn.), Whisper without Co-Prop., Whisper without LLMSyn., and the full method with all components enabled.

At 0.1% leakage, VAL baseline achieves a CRR of 16.74% on Enron and 18.13% on CC-News (Table 5). Although recovered anchors are accurate, overall recovery remains limited under such sparse observations. Incorporating Whisper's attack structure without AI-assisted signals increases CRR to 24.60% on Enron and 28.62% on CC-News, showing that global Hungarian matching over K/K' already extracts substantial signal from sparse co-occurrence statistics.

Introducing LLMSyn. alone (w/o Co-Prop.) improves CRR to 27.61% on Enron and 32.87% on CC-News, while introducing Co-Prop. alone (w/o LLMSyn.) raises it to 39.11% and 44.33%. Enabling both components yields full Whisper at 40.96% on Enron and 46.37% on CC-News, with a total improvement of 24.22 and 28.24 pp over the VAL baseline.

These gains reflect the combined effect of the two components in addressing co-occurrence sparsity. Co-Prop. provides broad-coverage semantic guidance for keyword pairs that never co-occur in the leaked documents, while LLMSyn. complements it by densifying local co-occurrence statistics within semantic clusters via generation of topically coherent auxiliary documents. This is supported by the marginal effects: adding LLMSyn. on top of Co-Prop. (w/o LLMSyn. $\rightarrow$ Full) yields improvements of 1.85 and 2.04 pp, while adding Co-Prop. on top of LLMSyn. (w/o Co-Prop. $\rightarrow$ Full) yields improvements of 13.35 and 13.50 pp, indicating uneven yet mutually reinforcing contributions under ultra-low leakage.

Semantic-fusion Strategies. Beyond *which* auxiliary signals help, we also study *how* the embedding signal should be integrated into the matching stage. Table 6 compares three mechanisms for incorporating semantic embeddings under different leakage levels on Enron and CC-News datasets. All variants share the same anchor stage and LLM-synthesized auxiliary corpus, differing only in how embedding similarity is integrated into the keyword association process.

Among them, *Direct propagation* performs two-hop diffusion over a kNN graph ($k = 20$, $\alpha = 0.3$) constructed from embedding similarity, where co-occurrence signals are propagated via symmetric normalization. *Late fusion* computes cosine similarity over both the empirical co-occurrence matrix K' and the embedding similarity matrix, and combines the resulting scores using $(1-\alpha)s_{\text{orig}} + \alpha s_{\text{ext}}$ ($\alpha = 0.3$) before Hungarian matching. Whisper integrates embedding similarity as a soft prior into K' prior to a single cosine similarity computation and Hungarian assignment. We evaluate all methods on Enron and CC-News under 0.1%, 0.3%, and 0.5% leakage, reporting CRR averaged over three runs.

Table 6: Semantic-fusion mechanism comparison.

Mechanism	Enron (%)			CC-News (%)		
	0.1%	0.3%	0.5%	0.1%	0.3%	0.5%
Direct propagation	30.66	51.75	61.77	31.18	61.42	75.29
Late fusion	37.29	60.31	69.40	38.62	69.86	81.79
Whisper (soft prior)	40.63	62.25	70.68	42.07	72.37	82.99

Table 7: Effect of global matching vs. greedy matching.

Leakage	Enron (%)			CC-News (%)		
	HMatch	GMatch	Δ CRR	HMatch	GMatch	Δ CRR
0.1%	41.25	30.22	+11.03	40.52	24.95	+15.56
0.2%	54.60	43.47	+11.12	61.85	40.34	+21.51
0.3%	62.89	51.10	+11.78	72.26	54.22	+18.04
0.4%	66.51	55.77	+10.74	83.56	69.40	+14.17
0.5%	70.45	60.95	+9.49	84.60	71.98	+12.62
0.6%	69.16	60.29	+8.87	89.94	80.60	+9.33
0.7%	73.81	65.51	+8.30	92.09	84.43	+7.66
0.8%	72.96	64.93	+8.03	92.15	84.99	+7.16
0.9%	79.13	71.40	+7.73	94.09	88.81	+5.28
1.0%	78.47	70.53	+7.93	95.03	89.70	+5.33

The results show consistent trends across all settings. Direct propagation performs the weakest, while late fusion consistently improves over it but remains below Whisper. The performance gap is more visible under lower leakage, where the empirical co-occurrence matrix is sparser. Overall, the results are consistent with the design choice of integrating semantic information earlier in the pipeline.

Global Ambiguity Resolution. The critical role of global ambiguity resolution is illustrated by comparing Hungarian matching against a greedy assignment. At 0.1% leakage, Hungarian matching achieves 41.25% CRR on Enron versus 30.22% for greedy selection, and 40.52% versus 24.95% on CC-News, an absolute improvement of 11–15% (Table 7). Table 7 further shows that while this gap narrows as leakage increases from 0.1% to 1%, Hungarian matching consistently outperforms greedy strategies, confirming that enforcing global one-to-one consistency mitigates error propagation inherent in local greedy decisions.

Figure 11 visualizes the trend at 0.1% leakage across datasets, showing that each component contributes incrementally: LLM densification improves coverage, semantic propagation enhances discrimination, and global matching ensures consistent resolution. Together, these components enable Whisper to remain effective even under extreme sparsity.

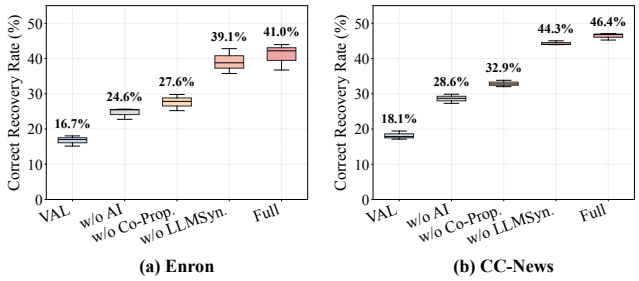


Figure 11: Incremental impact of Whisper components.

6.4 Robustness to Countermeasures

We evaluate the robustness of Whisper against representative leakage-mitigation countermeasures proposed in prior work [9, 22, 26]. Figure 12 summarizes the overall trends of Correct Recovery Rate under ultra-low leakage rates (0.1%-1.0%). We consider two widely adopted defenses that target different leakage dimensions: volume hiding, which pads all encrypted documents to a uniform size, and padding, which enforces response sizes to be multiples of n (we use $n = 500$) by injecting fake document-keyword associations.

The results show that volume hiding has a negligible impact on the effectiveness of Whisper. This behavior follows directly from the design of our anchor recovery mechanism. Document anchors in Whisper are characterized by the signature $\sigma(d) = (\mathrm{vol}(d), \deg(d))$, where the degree $\deg(d)$ is derived from the access-pattern matrix B and corresponds to the number of distinct keywords contained in a document. While volume hiding flattens document sizes and thus suppresses $\mathrm{vol}(d)$, it leaves the access-pattern matrix unchanged. Consequently, the degree-based structural signals used for document alignment remain intact, and the overall attack performance is largely unaffected.

In contrast, padding introduces a significant degradation in recovery accuracy for both attacks. Unlike volume hiding, padding does not merely suppress an existing leakage dimension but actively injects noise into the access-pattern matrix. By padding keyword occurrences to fixed multiples of n, it homogenizes keyword frequencies and creates artificial co-occurrences between unrelated keywords and documents. This dual effect simultaneously disrupts uniqueness of structural information and contaminates co-occurrence statistics.

Despite these challenges, Whisper still outperforms the baseline VAL. The performance gap stems from their fundamentally different disambiguation strategies. VAL relies primarily on local and structural information matching, making it highly sensitive to frequency poisoning and spurious co-occurrences. Whisper, although affected during the initial anchoring phase, leverages the densified semantic co-occurrence matrix as input to a global ambiguity resolution procedure. By formulating disambiguation as a maximum weighted bipartite matching problem and solving it using the Hungarian algorithm, Whisper enforces global consistency across candidate assignments. This global optimization mitigates the impact of local structural noise and prevents the cascading errors that cause greedy attacks to collapse under padding.

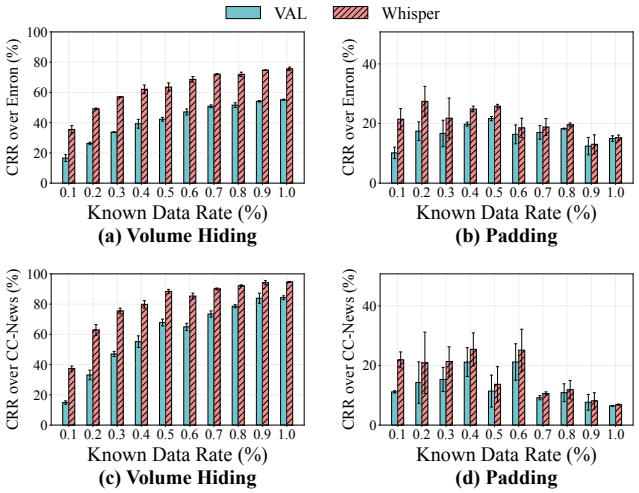


Figure 12: Attack effectiveness under countermeasures.

7 Discussion

Whisper shows that external semantic knowledge can substantially improve recovery under limited leakage. Its effectiveness, however, remains fundamentally constrained by the statistical alignment between auxiliary knowledge and the target database. As with prior leakage-abuse attacks, stronger overlap in keyword distributions and co-occurrence structures yields more reliable inference, while domain divergence reduces recovery accuracy. This is reflected in the performance gap between general-domain datasets (up to 95% CRR on Wiki and CC-News) and specialized corpora such as QEMU (69%), where domain-specific terminology is less effectively captured by general-purpose LLMs and embedding models.

This limitation also exists in the similar-data setting, where the adversary relies on statistical information derived from a distributionally similar corpus rather than exact leaked documents. In practice, such auxiliary data is often partial, outdated, or distributionally shifted. For example, an Enron-style adversary may only possess archived mailbox snapshots, while a CC-News-style adversary may rely on incomplete crawls whose topic distributions no longer align with the target collection. As a result, the co-occurrence statistics used by prior attacks can become noisy, reducing recovery reliability. This suggests that idealized same-corpus random splits, commonly used in prior evaluations, may overestimate the statistical alignment available in practice.

Extensions and Future Directions. Extending Whisper to similar-data settings is a natural next step. Because Whisper's anchor construction assumes exact document correspondence,

such an extension would require soft anchor scoring under partial matches, embedding-distance reweighting of auxiliary documents, and uncertainty-aware fusion of noisy co-occurrence signals. This extension could build on Whisper's AI-assisted components, with embedding-derived semantic priors smoothing sparse co-occurrence estimates and LLM-assisted expansion recovering semantic associations missing from incomplete auxiliary corpora.

Beyond adversarial knowledge assumptions, our evaluation also assumes static databases with fixed document collections and keyword universes. Extending Whisper to dynamic SSE would require update-aware modeling to track evolving co-occurrence statistics and maintain anchor stability under document churn. More broadly, promising directions include domain adaptation through fine-tuned embeddings or retrieval-augmented generation (RAG), multilingual model extensions, integration with active attacks, co-occurrence perturbation with formal privacy guarantees, and theoretical characterization of attack lower bounds and attack-defense tradeoffs.

8 Related Work

Except for the SSE schemes [14, 15, 36] based on expensive primitives, such as ORAM and PIR, most SSE schemes leak the access pattern, search, volume, and response size pattern (i.e., the size of each document). With some prior knowledge, passive attacks abuse the above leakages to recover users' queries. These attacks can be categorized as similar-data attacks [13, 23, 27, 28, 31] and known-data attacks [6, 9, 18, 22, 26, 37].

Known-data attacks. Islam et al. [18] introduced the first known-data attack (IKK), recovering queries using all documents and partial queries. Cash et al. [9] proposed Count, which recovers most queries without known queries. Blackstone et al. [6] achieve near-perfect recovery with fully known documents but fail (<10%) when only a small fraction is known. LEAP [26] recovers 50% of queries with 100% accuracy using just 1% of documents, while VAL [22] exploits leaked documents for enhanced recovery. All these attacks fundamentally rely on prior knowledge of the plaintext.

Similar-data attacks. These attacks exploit statistical correlations in query patterns rather than relying on known data. Freq [23] exploits search patterns and query frequencies, while SAP [27] leverages both search and volume patterns. Both achieve limited accuracy due to reliance on frequency information. Attacks exploiting access patterns are more effective: GraphM [31] models query recovery as a weighted graph matching problem solved via PATH [38] or Umeyama [35], and IHOP [28] combines query-keyword co-occurrence with frequency under a Markov correlation model, even circumventing frequency-smoothing defenses like PANCAKE [16]. RSA [13] achieves ∼85% accuracy but requires some known queries as input. Jigsaw [25] leverages the Zipfian distribution of query frequency and volume to prioritize high-frequency queries, improving recovery efficiency.

Table 8: Comparison of representative leakage-abuse attacks.

Attack	Leakage[§]	Known Data[‡]		Similar Data[‡]		Required[ι]	Reported[≀]
		Doc	Q	Doc	Freq		
IKK [18]	ap	●	◐	○	○	≥ 90% K	∼ 80%
Count [9]	ap, vp	●	○	○	○	∼ 100% K	∼ 99%
LEAP [26]	ap	◐	○	○	○	1–5% K	50–95%
VAL [22]	ap, vp	◐	○	○	○	1–5% K	50–95%
RSA [13]	ap	○	◐	●	○	∼ 40% S	∼ 85%
SAP [27]	vp, sp	○	○	●	●	∼ 50% S	∼ 50%
IHOP [28]	sp, ap	○	○	●	●	∼ 50% S	∼ 90%
Jigsaw [25]	vp, sp, ap	○	○	●	◐	∼ 50% S	∼ 90%
Whisper (Ours)	ap, vp	◐	○	○	○	0.1–1% K	36–95%

[§] ap = access pattern; vp = volume pattern; sp = search pattern.
[‡] ● strong dependence on known/similar data; ◐ partial use; ○ not used.
[ι] Auxiliary-data fraction at which the reported recovery is achieved; K = known-data regime, S = similar-data regime.
[≀] Representative recovery aligned to our Correct Recovery Rate (CRR) whenever directly comparable.

Table 8 summarizes representative passive leakage-abuse attacks across four dimensions: leakage types, auxiliary-data assumptions, required leakage regimes, and reported recovery performance. We align all results with our *Correct Recovery Rate* (CRR), defined as the fraction of correctly recovered queries. Prior work reports this metric under different names, including recovery rate, query recovery accuracy, and correct-match rate. As shown in Table 8, prior attacks cover a range of auxiliary-data assumptions. Earlier approaches such as IKK and Count require substantial known plaintext, while RSA, SAP, IHOP, and Jigsaw rely on auxiliary corpora drawn from distributions similar to the target dataset. LEAP and VAL extend this setting to lower-data known-plaintext regimes, but their performance degrades as the amount of available auxiliary data becomes extremely limited (Section 6). Whisper operates under the same passive leakage setting and low-data regime, while leveraging AI-synthesized co-occurrence and semantic signals to mitigate sparsity in the observed statistics.

LLMs for data synthesis. Large language models have emerged as powerful tools for synthetic data generation [7]. Carlini et al. [8] demonstrated that LLMs memorize and can leak training data, highlighting both their capacity to capture real-world distributions and associated privacy risks. A recent survey [24] systematically categorizes LLM-driven synthetic data generation techniques, including prompt-based generation, retrieval-augmented pipelines, and iterative self-refinement. Kim et al. [20] evaluated LLMs as synthetic data generators, while Kang et al. [19] conducted a large-scale empirical study on synthetic data scaling laws in LLM pre-training. These techniques have been successfully applied to augment training data in low-resource settings, reduce annotation costs, and improve model robustness. In the security domain, LLMs have been used for vulnerability discovery [29] and social engineering attacks [17]. To our knowledge, our work is the first to leverage LLMs for augmenting auxiliary

data in cryptographic leakage-abuse attacks, using synthetic document generation to enhance keyword co-occurrence matrices when the attacker's known data is limited.

9 Conclusion

This paper investigates passive known-data attacks against SSE under ultra-low leakage and makes a first step toward AI-augmented LAAs. We show that modern AI tools including LLMs and embeddings can be systematically exploited to compensate for sparse leakage by densifying keyword co-occurrence structures and injecting semantic regularities.

Extensive evaluation on six real-world datasets shows that Whisper consistently outperforms prior state-of-the-art attacks, achieving 2–2.5× higher recovery at 0.1% leakage (36%–49% vs. 17%–27%) and approaching 95% recovery at 1%. Even under extreme conditions with fewer than 0.1% leaked documents, Whisper attains non-trivial recovery and consistently outperforms existing attacks, albeit with higher variance. These results indicate that in the era of widely accessible AI models, ultra-low leakage is no longer a sufficient security assumption. They challenge existing SSE threat models and suggest that defenses must move beyond single-dimension leakage hiding toward protecting higher-order co-occurrence and semantic structures.

Acknowledgments

We are grateful to our shepherd and the anonymous reviewers for their valuable guidance and insightful comments. This work was supported by National Natural Science Foundation of China (No.U2336205, No.62302226, No.62502207, No.62302224), Basic Re-search Program of Jiangsu (No.BK20251448) and Yun-nan Provincial New R&D Institution Cultivation Project (No.202404BQ040148).

Ethical Considerations

This work studies passive leakage-abuse attacks against searchable symmetric encryption (SSE), showing that pre-trained embeddings and LLM-synthesized documents can amplify keyword recovery in ultra-low-leakage settings. We discuss the ethical implications in terms of stakeholders, impacts, misuse mitigations, and disclosure rationale.

Stakeholders. The relevant stakeholders include: (i) end users of SSE-protected services, such as encrypted email and encrypted enterprise search; (ii) service providers and enterprises that deploy SSE for privacy, compliance, or commercial reasons; (iii) SSE protocol designers and cryptographic researchers; (iv) regulators, auditors, and security operations teams assessing encrypted-search deployments; (v) providers of general-purpose LLMs and pre-trained embedding models, whose models may be repurposed as semantic priors; and (vi) the broader security community studying leakage-abuse attacks and defenses. Our results have direct implications for the first four groups and indirect implications for the latter two, as general-purpose AI models may serve as auxiliary resources in encrypted-search threat models.

Potential impacts. Our results show that, under the leakage profiles and datasets evaluated in this paper, even less than 1% known data can be sufficient to mount effective recovery attacks against SSE deployments that expose access and volume patterns. This finding can help defenders reassess assumptions about residual breaches, historical backups, and limited insider-access scenarios. It also provides systematic empirical evidence that general-purpose AI tools, including LLM synthesis and pre-trained embeddings, can materially amplify leakage-abuse attacks, giving the community concrete data for AI-assisted threat modeling. Releasing reproducibility artifacts further allows defenders and protocol designers to verify our claims and evaluate countermeasures, while the release is scoped to reproduce the paper's experiments rather than to target real-world deployments. At the same time, the methodology described in the paper could in principle inform malicious parties, and the availability of general-purpose AI systems may lower the barrier for reproducing attacks in settings previously considered safer. We believe the defensive benefit of publication outweighs this risk for the reasons discussed below.

Mitigations. All experiments are conducted locally on long-standing public research datasets (Enron, Apache Lucene mailing lists, Simple English Wikipedia, LKML, QEMU, and CC-News). We do not query any online SSE service or attempt to execute the attack against deployed systems. For the LLM synthesis component, seed documents are sanitized before submission to an external LLM service by removing email headers, quoted reply chains, contact information, and other potentially personally identifiable artifacts, reducing the risk of transmitting such content to a third party. Our release is limited to the general attack methodology and reproducibility scaffolding, and excludes turnkey exploitation tools and pre-trained attack artifacts targeted at any specific vendor or deployment. We also report the attack's empirical effectiveness under representative defenses (padding and volume hiding), giving defenders a concrete baseline for evaluating countermeasures.

Research and Publication Rationale. Leakage-abuse attacks against SSE are an established public research direction with more than a decade of prior work [9, 18, 22, 25, 26, 28]. Our contribution is not to introduce a new class of adversarial behavior, but to clarify two underexamined dimensions of this threat: the ultra-low-leakage regime and the use of general-purpose AI as an auxiliary signal. Withholding these results would not eliminate the threat, because resourceful adversaries could independently combine public leakage-abuse techniques with widely available AI models. In contrast, publication enables defenders, protocol designers, and auditors to

update their threat models and evaluate mitigations. Publishing therefore follows the standard practice in cryptographic and systems security research of disclosing attack results so that the community can respond.

Disclosure Scope. This work analyzes a paradigm-level property of the SSE leakage model, including access, volume, search-pattern, and keyword co-occurrence leakage, rather than an implementation-specific defect in any product. The attack does not rely on a zero-day flaw in any vendor implementation; the underlying leakage surface has been formally studied for over a decade and is openly discussed in the SSE literature. Consequently, vendor-by-vendor coordinated disclosure does not directly apply, and we have not identified or reported any exploitable flaw in a specific deployment. Our artifact is made publicly available to support reproducibility, with the released materials scoped to evaluating the attack and prospective countermeasures rather than targeting any specific real-world deployment.

Open Science

We release the artifact at https://doi.org/10.5281/zenodo.20355588. It contains the implementation of Whisper together with its greedy and late-fusion variants, as well as the VAL and LEAP baselines (`attacks/`), experiment scripts for reproducing results reported in the paper (`experiments/`), the LLM-assisted document synthesis pipeline (`synthesis/`), and dataset parsers for Enron, Lucene, Wikipedia, LKML, QEMU, and CC-News (`utils/`). A top-level README provides environment setup instructions, dataset sources, and exact commands for reproducing the main figures and tables.

References

[1] Linux kernel mailing list. https://lkml.org/. Emails from October 2025–January 2026.

[2] QEMU development mailing list. https://lists.gnu.org/archive/mbox/qemu-devel/. Emails from September 2025 – January 2026.

[3] Simple English Wikipedia dump. https://dumps.wikimedia.org/simplewiki/. Accessed: 2025.

[4] Apache Foundation. Mail archives of lucene. https://mail-archives.apache.org/mod_mbox/#lucene, 1999.

[5] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc., 2009.

[6] Laura Blackstone, Seny Kamara, and Tarik Moataz. Revisiting leakage abuse attacks. In *Proceedings 2020 Network and Distributed System Security Symposium*. Internet Society, 2020.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[8] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *USENIX Security Symposium*, pages 2633–2650, 2021.

[9] David Cash, Paul Grubbs, Jason Perry, and Thomas Ristenpart. Leakage-abuse attacks against searchable encryption. In *ACM Conference on Computer and Communications Security (CCS)*, pages 668–679, 2015.

[10] William W. Cohen. Enron email dataset. https://www.cs.cmu.edu/~enron/, 2015.

[11] Common Crawl Foundation. Common crawl news dataset (CC-NEWS). https://commoncrawl.org/blog/news-dataset-available. WARC archives from 2025–2026.

[12] Reza Curtmola, Juan Garay, Seny Kamara, and Rafail Ostrovsky. Searchable symmetric encryption: improved definitions and efficient constructions. In *ACM Conference on Computer and Communications Security (CCS)*, pages 79–88, 2006.

[13] Marc Damie, Florian Hahn, and Andreas Peter. A highly accurate query-recovery attack against searchable encryption using non-indexed documents. In *USENIX Security Symposium*, pages 143–160, 2021.

[14] Ioannis Demertzis, Dimitrios Papadopoulos, Charalampos Papamanthou, and Saurabh Shintre. SEAL: Attack mitigation for encrypted databases via adjustable leakage. In *29th USENIX security symposium (USENIX Security 20)*, pages 2433–2450, 2020.

[15] Sanjam Garg, Payman Mohassel, and Charalampos Papamanthou. Tworam: Efficient oblivious ram in two rounds with applications to searchable encryption. In *Annual International Cryptology Conference*, pages 563–592. Springer, 2016.

[16] Paul Grubbs, Anurag Khandelwal, Marie-Sarah Lacharité, Lloyd Brown, Lucy Li, Rachit Agarwal, and Thomas Ristenpart. Pancake: Frequency smoothing for encrypted data stores. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 2451–2468, 2020.

[17] Julian Hazell. Spear phishing with large language models. *arXiv preprint arXiv:2305.06972*, 2023.

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

[19] Feiyang Kang, Newsha Ardalani, Michael Kuchnik, Youssef Emad, Mostafa Elhoushi, Shubhabrata Sengupta, Shang-Wen Li, Ramya Raghavendra, Ruoxi Jia, and Carole-Jean Wu. Demystifying synthetic data in llm pre-training: A systematic study of scaling laws, benefits, and pitfalls. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 10739–10758, 2025.

[20] Seungone Kim, Juyoung Suk, Xiang Yue, Vijay Viswanathan, Seongyun Lee, Yizhong Wang, Kiril Gashteovski, Carolin Lawrence, Sean Welleck, and Graham Neubig. Evaluating language models as synthetic data generators. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6385–6403, 2025.

[21] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[22] Steven Lambregts, Huanhuan Chen, Jianting Ning, and Kaitai Liang. VAL: Volume and access pattern leakage-abuse attack with leaked documents. In *European Symposium on Research in Computer Security (ESORICS)*, pages 653–676. Springer, 2022.

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

[24] Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. On llms-driven synthetic data generation, curation, and evaluation: A survey. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 11065–11082, 2024.

[25] Hao Nie, Wei Wang, Peng Xu, Xianglong Zhang, Laurence T. Yang, and Kaitai Liang. Query recovery from easy to hard: Jigsaw attack against SSE. In *USENIX Security Symposium*, pages 2599–2616, 2024.

[26] Jianting Ning, Xinyi Huang, Geong Sen Poh, Jiaming Yuan, Yingjiu Li, Jian Weng, and Robert H. Deng. LEAP: Leakage-abuse attack on efficiently deployable, efficiently searchable encryption with partially known dataset. In *ACM Conference on Computer and Communications Security (CCS)*, pages 2307–2320, 2021.

[27] Simon Oya and Florian Kerschbaum. Hiding the access pattern is not enough: Exploiting search pattern leakage in searchable encryption. In *USENIX Security Symposium*, pages 127–142, 2021.

[28] Simon Oya and Florian Kerschbaum. IHOP: Improved statistical query recovery against searchable symmetric encryption through quadratic optimization. In *USENIX Security Symposium*, pages 2407–2424, 2022.

[29] Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt, and Ramesh Karri. Asleep at the keyboard? assessing the security of GitHub Copilot's code contributions. In *IEEE Symposium on Security and Privacy (S&P)*, pages 754–768, 2022.

[30] Martin F Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[31] David Pouliot and Charles V. Wright. The shadow nemesis: Inference attacks on efficiently deployable, efficiently searchable encryption. In *ACM Conference on Computer and Communications Security (CCS)*, pages 1341–1352, 2016.

[32] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3982–3992, 2019.

[33] Dawn Xiaoding Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *IEEE Symposium on Security and Privacy (S&P)*, pages 44–55. IEEE, 2000.

[34] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[35] S. Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE transactions on pattern analysis and machine intelligence*, 10(5):695–703, 1988.

[36] Zhiqiang Wu and Rui Li. Obi: a multi-path oblivious ram for forward-and-backward-secure searchable encryption. In *NDSS*, 2023.

[37] Lei Xu, Huayi Duan, Anxin Zhou, Xingliang Yuan, and Cong Wang. Interpreting and mitigating leakage-abuse attacks in searchable symmetric encryption. *IEEE Transactions on Information Forensics and Security*, 16:5310–5325, 2021.

[38] Mikhail Zaslavskiy, Francis Bach, and Jean-Philippe Vert. A path following algorithm for the graph matching

problem. *IEEE transactions on pattern analysis and machine intelligence*, 31(12):2227–2242, 2009.

A Impact of Keyword Universe

We evaluate the effectiveness of Whisper and VAL under varying sizes of the keyword universe to assess their robustness when queries cover only a subset of all keywords. As shown in Table 9, we report the Correct Recovery Rate (CRR) on the Enron dataset for three keyword universe sizes ($|\mathcal{W}| \in \{1000, 3000, 5000\}$) and leakage rates ranging from 0.1% to 1.0%.

Table 9: CRR (%) of query under different size of keyword universe on Enron dataset: VAL vs. Whisper.

Leakage	$\|\mathcal{W}\| = 1000$		$\|\mathcal{W}\| = 3000$		$\|\mathcal{W}\| = 5000$	
	VAL	Whisper	VAL	Whisper	VAL	Whisper
0.1%	27.40	61.24	18.72	45.25	16.74	41.90
0.2%	53.84	82.56	34.11	60.04	30.37	54.81
0.3%	68.66	91.82	43.49	70.75	38.32	62.89
0.4%	83.45	96.39	50.65	75.80	42.92	66.79
0.5%	89.83	98.00	56.74	79.90	48.13	70.60
0.6%	92.59	98.39	57.78	79.92	47.59	69.31
0.7%	94.43	98.97	64.46	85.36	52.62	73.77
0.8%	97.06	99.70	65.03	84.41	52.90	72.93
0.9%	99.20	100.0	71.02	89.36	58.15	79.26
1.0%	99.30	99.80	71.81	89.18	58.31	78.49

Across all tested configurations, Whisper consistently outperforms VAL. When $|\mathcal{W}| = 1000$, Whisper achieves a CRR of 61.24% at 0.1% leakage and reaches perfect recovery at 0.9% leakage, while VAL attains only 27.40% and peaks at 99.30%, respectively.

As the size of the keyword universe increases, both attacks exhibit performance degradation due to partial observation of the query-document incidence matrix, which results in systematically underestimated document degrees and weaker anchor signals. Nevertheless, Whisper maintains a clear advantage. For instance, at $|\mathcal{W}| = 5000$ with 1.0% leakage, Whisper achieves a CRR of 78.49%, compared to 58.31% for VAL, an increase of about 20%.

These results indicate that although limited coverage of keyword universe affects all known-data attacks, Whisper remains effective under more realistic query workloads, thus alleviating concerns about reliance on idealized full-coverage assumptions.

B Comparison with Other Baselines

To further validate the effectiveness of Whisper, we include additional comparisons with representative leakage-abuse attacks under the known-data setting on the Enron dataset. We consider two baselines: (i) Count [9], one of the earliest known-data attacks that relies solely on volume information, and (ii) LEAP [26], a widely studied attack that exploits document frequency. All methods are evaluated using *Correct Recovery Rate* (CRR) of query recovery.

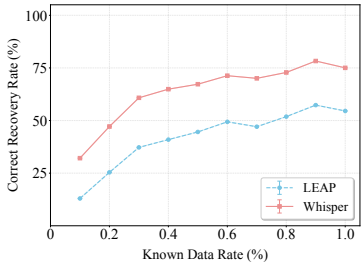


Figure 13: CRR of query: Whisper vs. LEAP.

Comparison with Count. Under the known-data setting, the Count attack exhibits limited recovery capability. With 1% known data, Count fails to recover any queries, yielding a CRR of 0%. Even when the known data rate increases to 75%, the CRR remains below 7%. A noticeable improvement is observed only when the known data rate approaches very high levels, reflecting Count's strong dependence on accurate volume matching. Overall, Count performs substantially worse than Whisper, highlighting the limitations of purely volume-based attacks when empirical evidence is sparse.

Comparison with LEAP. Figure 13 reports the CRR of query recovery on the Enron dataset as the known data rate varies from 0.1% to 1.0%. Across all evaluated leakage rates, Whisper consistently outperforms LEAP. At 0.1% known data, Whisper achieves approximately 32.12% CRR, compared to 12% achieved by LEAP, corresponding to a 2.7× improvement. This performance gap persists across the entire range: at 1.0% known data, Whisper reaches approximately 75.04% CRR, while LEAP saturates at around 55%. The advantage of Whisper stems from its semantic co-occurrence densification mechanism, which provides richer disambiguation signals, particularly under sparse known-data conditions.