Bayesian Leakage Analysis

A Framework for Analyzing Leakage in Cryptography

Zachary Espiritu[1], Seny Kamara[1,2] and Tarik Moataz[1]

[1] MongoDB, New York, United States
[2] Brown University, Providence, United States

Abstract. We introduce a framework based on Bayesian statistical inference for analyzing leakage and its vulnerability to inference attacks. Our framework naturally integrates auxiliary information, defines a notion of adversarial advantage, and provides information-theoretic measures that capture the security of leakage patterns against both full and functional recovery attacks.

We present two main theorems that bound the advantage of powerful inference techniques: the maximum a posteriori (MAP), the maximum likelihood estimate (MLE) and the MAP test. Specifically, we show that the advantage of these methods is exponentially bounded by new entropy measures that capture the susceptibility of leakage patterns to inference.

To demonstrate the applicability of our framework, we design and implement an automated leakage attack engine, `Bayle`, which leverages a novel inference algorithm that efficiently computes MAP estimates for a large class of i.i.d. leakage models. These models include, for example, query equality, the combination of query equality and volume, and leakage patterns arising from naive conjunctions.

Keywords: Encrypted search · leakage · leakage attacks

1 Introduction

Sub-linear encrypted search algorithms (ESA) are highly-efficient search algorithms that can be executed on end-to-end encrypted data. ESAs are the core building block in the design of a variety of end-to-end encrypted systems including encrypted cloud storage [CJJ+14, CJJ+14, PKV+14, FVK+15] and encrypted databases [APRZB11, KM18, KMZZ20, ZKMZ21, KMPQ21, Mon]. Sub-linear ESAs can be designed based on a variety of cryptographic primitives including structured encryption (STE), oblivious RAM (ORAM) and property-preserving encryption (PPE). Intuitively, an ESA should reveal no partial information about the data and/or queries but all sub-linear ESAs leak some information. This leakage is typically captured with a *leakage profile* that formally and precisely describes what the scheme reveals. While leakage profiles have proven to be an important conceptual and analytical tool, they are purely descriptive.explanatory value.

The presence of leakage in ESAs has motivated several complimentary research agendas: *leakage cryptanalysis* which focuses on the design of attacks that try to exploit various leakage profiles in order to recover information about the data and/or queries [IKK12, NKW15, CGPR15, BKM20, KKNN16, LMP18, GLMP18, GLMP19, KPT20, KPT21]; *leakage suppression* which focuses on techniques to design low-leakage, zero-leakage and subliminal ESAs [KMO18, KM19, GKM21, PPYY19, APP+21]; and *leakage quantification* which focuses on quantifying the information revealed by a given leakage profile [JS19, JPS21, KMPP22].

E-mail: zachary.espiritu@mongodb.com (Zachary Espiritu), seny.kamara@mongodb.com (Seny Kamara), tarik.moataz@mongodb.com (Tarik Moataz)

Leakage of cryptographic primitives. While the focus of this work is on the leakage produced by sub-linear ESAs, we note that leakage is pervasive throughout cryptography.[1] This includes perfectly-secure [Sha49] and CPA-secure encryption schemes [GM82], which reveal the length of messages; oblivious RAM (ORAM) [GO96] and private information retrieval (PIR) [CGKS95], which reveal the number of items stored and the number of accesses; anonymous communication protocols [Cha81], which reveal certain traffic patterns; differentially private mechanisms [DMNS06], which reveal global information about the dataset; secure multi-party computation (MPC) protocols [Yao82, GMW87] which reveal information about honest party inputs that is a function of the output and obfuscation [BGI+01, SW14], which reveals certain characteristics of the circuit/program. The foundations of cryptography, however, do not provide a principled way to study such leakage, e.g., there is no established framework to study seemingly fundamental questions like

> *can revealing the length, order or location of a secret lead to a security violation and, if so, under what conditions?*

Leakage in definitions. Leakage appears in security definitions and it is typically handled using one of two approaches:

- *(implicit)* leakage is implicitly modeled (e.g., by constraining the choices of the adversary in game-based definitions or by using backchannels in simulation-based definitions) and taken as a natural limitation of the primitive and is not considered further. For example, this is the case in the security definitions of encryption, MPC, ORAM, and PIR as mentioned above.

- *(explicit)* leakage is explicitly modeled (e.g., as a leakage function) and no assumption is made on whether it is "natural" or not.

The main limitation of the implicit approach is that the line between natural and unnatural leakage is mostly arbitrary since the exploitability of leakage depends on the context in which it is revealed. For example, in the case of encryption, standard security definitions guarantee that all partial information about the message beyond its length and what the adversary already knows is hidden. This length leakage seems acceptable when transmitting fixed-length messages but can become problematic when transmitting variable-length messages and, in particular, compressed messages [Kel02, WBMM07]. Similarly, accessing an ORAM in a way that is indepenent of a secret is acceptable but the number of ORAM accesses can become an issue if it depends on a secret. Unlike the implicit approach, the explicit approach does not consider any leakage as natural and instead examines its implications in specific settings. Note that, in this approach, leakage itself is elevated to being an object for cryptographic study.

Since its introduction in [CGKO06, CK10], the explicit approach has become standard in encrypted search and is increasingly being applied in other areas of cryptography. While it has focused cryptographers' attention on leakage and driven advances in leakage cryptanalysis, it is only meant to make it explicit. As such, a crucial foundational component that has been absent is a formal and theoretical framework for systematically studying leakage and its susceptibility to leakage attacks. Developing such a framework has been an open problem for the last twenty years and has been a central topic of discussion in encrypted search since the introduction of the explicit approach.

[1] Here, we are referring to leakage that results from the primitive itself in the model in which it is proven secure, as opposed to, e.g., side-channel leakage or leakage about keying material as in leakage-resilient cryptography.

Auxiliary information. One of the main challenges in developing a useful "theory of leakage" is that leakage itself is not the only component of a leakage attack. In fact, as critical as leakage is, in many attacks, it is the combination of leakage and auxiliary information that results in information being extracted. In other words, for a leakage framework to be interesting and applicable in practice, it must not only capture leakage but must also integrate auxiliary information and, crucially, model the interaction between the two in order to accurately evaluate the susceptibility to attack.

An overview of our framework. In this work, we propose a new theoretical framework to analyze leakage and its vulnerability to inference attacks. Our framework offers several key advantages: (1) a natural way to incorporate auxiliary information; (2) a definition of advantage that quantifies an adversary's success probability in exploiting a leakage profile using auxiliary information; (3) a set of information-theoretic measures that formalize the characteristics of leakage that determine the hardness of full and partial recovery; and (4) an automated method—subject to computational constraints—to attack a common class of leakage patterns.

We begin by modeling leakage profiles and the design of leakage attacks as a concrete instantiation of the following statistical inference problem. Let $\mathbf{X} = (\mathbf{S}, \mathbf{H}, \mathbf{L})$ be a joint random variable consisting of multivariate variables $\mathbf{S}$ that output a secret, $\mathbf{H}$ that output hidden/unobservable elements and $\mathbf{L}$ that output leakage. Additionally, let α be a probability distribution over the secret space that captures the adversary's auxiliary information about the secret. Throughout, we refer to $\mathbf{X}$ as a *leakage model*. A leakage attack is an inference/estimation algorithm that, given an instantiation $\boldsymbol{\ell}$ of $\mathbf{L}$ and α in explicit form, returns information about the instantiation $\mathbf{s}$ of the secret variables $\mathbf{S}$ that led to the observed leakage. If the goal is to recover these instantiations in their entirety, the attack is a *full recovery attack* and if the goal is to recover some function of the instantiations, then it is a *functional recovery attack*. If the objective is only to approximate the instantiations, the attack is an *approximation attack*.

In the case of encrypted search, the secret variables usually consist of query variables $\mathbf{Q}$ and/or data variables $\mathbf{D}$ that produce a query sequence $\boldsymbol{q}$ and a dataset $\boldsymbol{d}$, respectively. If the target of the attack is the query sequence, then we refer to the attack as a *query recovery attack*; whereas if the target is the dataset, we refer to it a *data recovery attack*.

Advantage. In the literature on leakage cryptanalysis, leakage attacks are usually evaluated by reporting their *recovery rate*, which is the fraction of secrets from a target dataset that the attack recovers correctly. In Section 4, we discuss in detail why the recovery rate is a misleading metric but the main issue is that, if an attack leverages auxiliary information, then its success should be compared against a trivial attack that only leverages this auxiliary information. Without doing so, one cannot rule out the possibility that the attack completely ignores the leakage and extracts information directly from the auxiliary data. To address this we formalize a notion of advantage that works as follows. For a leakage model $\mathbf{X}$, we first define an inference experiment $\mathbf{Inf}$ where the adversary is given some observed leakage $\boldsymbol{\ell}$ such that $(\mathbf{s}, \boldsymbol{h}, \boldsymbol{\ell}) \leftarrow \mathbf{X}$, and the description of an auxiliary probability distribution α over the secret space. In the case of full recovery, for example, the experiment succeeds if the adversary's estimate $\hat{\mathbf{s}}$ of the secret is equal to the secret $\mathbf{s}$. We then define an experiment $\mathbf{Guess}$ which samples a secret $(\mathbf{s}, \boldsymbol{h}, \boldsymbol{\ell}) \leftarrow \mathbf{X}$, sets the estimate $\hat{\mathbf{s}}$ to be a mode of α's support and succeeds if $\hat{\mathbf{s}} = \mathbf{s}$. We define the advantage of the adversary as the absolute difference between the probabilities that these experiments succeed. The case of functional recovery is defined based on a natural variation of these experiments.

Adversaries. We focus on three adversaries based on standard statistical inference methods. In the case of full recovery, we consider the MAP adversary, $\mathcal{A}_{\mathsf{map}}$, which computes the maximum a-posteriori (MAP) estimate of the secret and the MLE adversary, $\mathcal{A}_{\mathsf{mle}}$, which computes the maximum likelihood estimate (MLE). The MAP estimate of a random variable $\mathbf{S}$ is the instantiation of the variable that maximizes the posterior probability; that is, the secret that maximizes the probability over the secret space $\mathbb{S}$ conditioned on the leakage variables being instantiated by $\boldsymbol{\ell}$. The MLE of a random variable $\mathbf{S}$ is the instantiation that maximizes the likelihood of the observed value; that is, the secret that assigns the most probability to the observed leakage. For functional recovery, we consider the MAP test adversary, $\mathcal{A}_{\mathsf{tst}}$, which computes the MAP test over a recovery function f. The MAP test outputs the element in the co-domain of f with the highest-probability pre-image with respect to the posterior distribution. We call this the MAP test adversary because it corresponds to the MAP test from Bayesian hypothesis testing, where the hypotheses are of the form $H_y : f(\mathbf{s}) = y$, with $f : \mathbb{S} \to \mathbb{Y}$.

Bounds for full recovery. The goal of leakage analysis is to better understand the conditions under which a leakage profile is hard to exploit. With our framework in place, we can reframe this question precisely as follows: given a leakage model $\mathbf{X}$, can one bound the advantage of the adversaries $\mathcal{A}_{\mathsf{map}}$, $\mathcal{A}_{\mathsf{mle}}$ and $\mathcal{A}_{\mathsf{tst}}$ with respect to an auxiliary distribution α. We address this question for the case of full recovery in our main results of Section 4. The first characterizes the advantage of $\mathcal{A}_{\mathsf{map}}$ and the second the advantage of $\mathcal{A}_{\mathsf{mle}}$ as a function of three information-theoretic measures we introduce:

- the *leakage entropy* quantifies the indistinguishability of the secrets with respect to the leakage they produce and the a-priori beliefs the auxiliary distribution assigns to them. An increase in leakage entropy decreases the divergence between the leakage model's likelihood functions (i.e., the probability distributions over observable leakages conditioned on a secret), introducing ambiguity in the observed leakage that confuses statistical estimation;
- the *singular entropy* measures how susceptible a model is to *direct inversion*, where an unbounded adversary can trivially recover the secret because secrets and leakages are in a one-to-one correspondence;
- the *modal entropy* measures how well the adversary's strongest beliefs about the secrets (according to the auxiliary distribution) align with the true distribution of secrets.

Based on these measures we present in Section 4 a Theorem showing that the advantage of $\mathcal{A}_{\mathsf{map}}$ against a leakage model $\mathbf{X}$ is at most

$$2^{-\Lambda(\mathbf{X},\alpha)} + 2^{-\Sigma(\mathbf{X})} + 2^{-\chi(p_{\mathbf{S}},\alpha)},$$

where $\Lambda(\mathbf{X})$ is the leakage entropy of $\mathbf{X}$ with respect to α, $\Sigma(\mathbf{X})$ is the singular entropy of $\mathbf{X}$, and $\chi(p_{\mathbf{S}}, \alpha)$ is the modal entropy of $p_{\mathbf{S}}$ with respect to α. We then show a Corollary that proves that the advantage of $\mathcal{A}_{\mathsf{mle}}$ against a leakage model $\mathbf{X}$ is at most

$$2^{-\Lambda(\mathbf{X},u_{\mathbb{S}})} + 2^{-\Sigma(\mathbf{X})} + \frac{1}{\#\mathbb{S}},$$

where $u_{\mathbb{S}}$ is the uniform distribution over the secret space.

In Section 7, we use our Theorem to study the case of query equality leakage over Zipf-distributed queries with respect to Zipf-distributed auxiliary distributions with the same parameter. We show that, in this case, the advantage of $\mathcal{A}_{\mathsf{map}}$ is at most

$$H_{m,\gamma}^{-n} + \frac{\log H_{m,\gamma}}{\log(m^{\gamma} \cdot H_{m,\gamma})},$$

where m is the size of the query space and $H_{m,\gamma}$ is the generalized Harmonic number. The above bound can be approximated by

$$H_{m,\gamma}^{-n} + \frac{\log(\log m + 1)}{\gamma \cdot \log(m)}.$$

For $\gamma = 1$ or $\gamma = 2$, this bound is $O(\log \log m / \log m)$, where m is the size of the query space. This is somewhat surprising because it shows that even when the adversary has an accurate auxiliary distribution, the attack can still underperform. We also study the case of Zipf-distributed queries with respect to Zipf-distributed auxiliary distributions with *distinct* parameters $\gamma \neq \gamma'$. In particular, we show that the advantage of $\mathcal{A}_{\mathsf{map}}$ is at most

$$H_{m,\gamma}^{-n} + \frac{\log H_{m,\gamma'}}{\log(m^{\gamma'} \cdot H_{m,\gamma'})}.$$

Finally, we also study the case of Zipf-distributed queries with respect to Zipf-distributed auxiliary distributions with distinct parameters $\gamma \neq \gamma'$ and distinct underlying permutations. Note that for a Zipf distribution, one has to implicitly define a bijection π that maps a query to a rank. Assuming that π_q and π_a are the underlying bijections of the query and auxiliary distributions, respectively, we show that the advantage of $\mathcal{A}_{\mathsf{map}}$ is at most

$$\left(\pi_q\left(\pi_a^{-1}(1)\right)^{-\gamma} \cdot H_{m,\gamma}\right)^{-n} + \frac{\log H_{m,\gamma'}}{\log(m^{\gamma'} \cdot H_{m,\gamma'})}.$$

The worst case occurs when the query with the highest mass in the query distribution is the query with the smallest mass in the auxiliary distribution. In this case, $(\pi_q(\pi_a^{-1}(1))^{-\gamma} \cdot H_{m,\gamma})^{-n} = m$.

Bounds for functional recovery. In our second Theorem, we examine the case of functional recovery, where an adversary aims to compute a function f of the secret $\mathbf{s}$. We present a bound that characterizes the advantage of $\mathcal{A}_{\mathsf{tst}}$ as a function of more complex variants of the information-theoretic measures discussed above. Specifically, we show that the advantage of $\mathcal{A}_{\mathsf{tst}}$ against a leakage model $\mathbf{X}$ is at most

$$2^{-\Gamma(\mathbf{X},f,\alpha)} + 2^{-\Phi(\mathbf{X},f)} + 2^{-\Psi(f,p_{\mathbf{S}},\alpha)}.$$

Here, $\Gamma(\mathbf{X}, f, \alpha)$ is a measure we call the *functional leakage entropy* of $\mathbf{X}$, which quantifies the difficulty of estimating $f(\mathbf{s})$ given the observed leakage. This measure depends on the number and distribution of secrets that f maps to an incorrect recovery value $y \neq f(\mathbf{s})$. Additionally, $\Phi(\mathbf{X}, f)$ is a measure we refer to as the *functional singular entropy*, which quantifies how susceptible a model is to the direct estimation of the recovery value and $\Psi(f, p_{\mathbf{S}}, \alpha)$ is what we refer to as the *functional modal entropy*, which measures how well the auxiliary distribution's strongest beliefs align with function values under the true distribution. As mentioned, these information measures are complex, and we were unable to compute them for specific leakage patterns and common recovery functions. Nevertheless, we believe that the bound remains a valuable contribution and hope that it will inspire future work, potentially leading to bounds based on simpler measures.

Automated leakage attacks. In addition to naturally integrating auxiliary information, our framework supports the use of statistical and machine learning algorithms. This capability allows us to construct *leakage attack engines*, which we define as tools that can automatically attack common classes of leakage models without requiring the design of custom attacks. In this work, we demonstrate the feasibility of this idea by designing and implementing an attack engine called `Bayle` that targets a wide range of leakage models, including the i.i.d. variants of the most common leakages: the query equality pattern, the combination of query equality and volume and two conjunctive leakages.

A new inference algorithm. Our first step in designing our engine is to restrict our attention to leakage models that can be formalized as a subclass of Bayesian Networks (BNs), which are probabilistic graphical models widely used in statistics and machine learning. All the leakage profiles we are aware of can be modeled as BNs, though it is possible that more complex profiles may emerge in the future.

Roughly speaking, a BN is a graph whose vertices represent the model's random variables and edges represent conditional dependencies. One can think of a BN as a compact representation of a joint distribution that can be leveraged to efficiently compute a variety of probability distributions associated with the model. MAP estimates on certain BNs can be computed efficiently using the Variable Elimination (VE) algorithm [LS88] which has a running time of $O(n \cdot d^{w+1})$ and a space complexity of $O(d^{w+1})$, where d is the maximum domain size of the random variables and w is the treewidth of the BN. While BNs in machine learning tend to have small domains, this is unfortunately not the case in our setting.

To address this, we first observe that the BNs of many leakage profiles share a similar structure. We refer to these networks as *hidden function networks* (HFNs) and formalize them in Section 8.1. At a high level, these leakage models generate leakage sequences $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_n)$ such that each observed leakage $\ell_i = (h_1(s_i), \ldots, h_k(s_i))$ results from evaluating a set of hidden functions $\boldsymbol{h} = (h_1, \ldots, h_k)$ on the secrets. We then design a new MAP inference algorithm specifically for HFNs over independent and identically distributed (i.i.d.) secret variables. When at least one of the hidden functions is bijective, our algorithm is considerably faster than running VE on HFNs. Although the comparison is not entirely fair since the algorithms were implemented in different programming languages, our `Python` implementation of VE took 2 hours to estimate a sequence of 10 i.i.d. secrets from query equality leakage over a query space of size 9, whereas a `Julia` implementation of our custom algorithm took 31.3 ms.

Empirical analysis of leakage patterns. We used our engine to analyze the HFNs of several common leakage patterns, including query equality, the combined query equality and volume, and leakage patterns resulting from naive conjunctions. We focused on these because they are among the most prevalent and difficult to suppress. Additionally, we examined the leakages from naive conjunctions to determine whether the common belief that they leak excessively holds true. In our experiments, we used a Zipf distribution for the query distributions and a uniform distribution for volume. Overall, our experiments show that $\mathcal{A}_{\mathsf{map}}$ has a higher advantage in query recovery when its auxiliary distribution is the same as or very close to the query distribution, when the query sequences are longer, and when the query space is smaller. In the case of naive conjunctions, the attack achieved a very high advantage.

Connections and new directions. In addition to providing theoretical and algorithmic techniques to study leakage, our framework naturally connects the analysis of leakage to other areas of computer science and mathematics. Through these connections, we hope that techniques from these fields can be leveraged to analyze leakage and that leakage, in turn, can motivate new research problems within these areas. These connections are discussed in more detail in Section 4, but at a high level, they are as follows:

- *(Bayesian statistics)* our leakage analysis framework is based on Bayesian statistics, a powerful statistical paradigm with strong theoretical underpinnings. Conducting leakage analysis within the Bayesian framework allows us to benefit from many statistical results. For example, one can show using a fairly simple decision-theoretic argument that MAP estimates are optimal in the sense that they minimize the expected zero-one loss. However, this argument does not trivially extend to our setting since the MAP estimate is computed using an auxiliary distribution. Therefore,

it would be interesting to show (or disprove) that the MAP adversary is optimal in the same sense either for all auxiliary distributions or for some class of auxiliary distributions.[2]

- *(information theory)* the notions of entropy we introduce for the study of leakage are connected to the Kullback-Leibler (KL) divergence and, in some cases, to collision entropy. It would be interesting, however, to study these measures further and establish bounds that formally connect them to standard information-theoretic concepts.

- *(computational statistics and machine learning)* as mentioned above, our inference algorithm for HFNs has two main limitations: (1) it relies on the secret variables being i.i.d.; and (2) its running time is exponential in the domain size of the variables. A natural question then is how to design inference algorithms for HFNs (and possibly other cryptographically relevant Bayesian Networks) that are efficient without relying on the i.i.d. assumption. Another interesting direction is to consider *approximate attacks*, which could be implemented using approximate inference algorithms such as *variational inference* [JGJS99] or Markov Chain Monte Carlo (MCMC) methods [MRR+53, Has70, RC04].

- *(average-case hardness)* our framework and the bounds we prove are information-theoretic, meaning they bound the advantage of computationally unbounded adversaries. A natural question then is whether stronger bounds exist for computationally bounded adversaries. Specific examples of leakage profiles that are computationally secure were first presented in [KM19], but it would be interesting to obtain general results about the limits of computationally-bounded leakage attacks. We note that this question is directly related to an extensive line of work in average-case complexity theory concerning the computational hardness of Bayesian inference [Bar16, Hop18, BBH18, BHK+19, BB20, BBH+21]. These works prove lower bounds on various statistical inference problems like the planted clique problem in restricted models of computation, such as the Sum-of-Squares (SOS) hierarchy [Las01, Par03] and the Statistical Query (SQ) model [Kea93]. An important research direction motivated by our work is to explore how these hardness assumptions can be used to study the limits of computationally-bounded leakage attacks.

- *(Fourier analysis of Boolean functions)* our notion of functional leakage entropy is defined for a given Boolean function f. It would be interesting to study the functional leakage entropy of classes of recovery functions, however. Such results may be obtainable by using ideas from the Fourier analysis of Boolean functions, which allows us to reason about and approximate Boolean functions in a structured way.

- *(cryptography)* though our framework was motivated by the study of leakage in encrypted search, it can be used to analyze leakage throughout cryptography since leakage appears in many cryptographic settings. Some examples include secure multi-party computation [FIM+06, MF06, HKE12, KMRR15], private set intersection [GRR19], topology-hiding computation [BBMM18], differential privacy [DMNS06] and anonymization networks.

2 Related Work

We review related work that proposes leakage analysis frameworks. We note that our focus is on comparing the *frameworks* that are described in these works and not the specific

[2] The standard argument holds if the auxiliary distribution is the same as the secret distribution.

results proved using the frameworks.

Biased coin game. In [WP17], Wright and Pouliot propose a framework to study full data recovery attacks against the leakage of deterministic (DTE) and order-revealing encryption (ORE). At a high level, their approach consists of reducing the problem of recovering DTE- and ORE-encrypted data to winning two games the authors call the biased coin game (BCG) and the loaded dice game (LDG). In the (m, n)-BCG, a challenger holds m biased coins each of which lands heads with probability p_i. The challenger then samples a coin according to a prior distribution π and tosses that coin n times. It then provides its prior distribution over coins, the coin probabilities $(p_1, \ldots, p_m)$ and the results of the n coin tosses to an adversary whose goal is to guess which coin was chosen. The (m, n, d)-LDG is a generalization of the BCG to d-sided die. The authors then show how winning the BCG leads to a full data recovery attack on DTE and how winning the LDG leads to a full data recovery attack on ORE.

Quantitative information flow. In a pair of works, Jurado and Smith [JS19] and later Jurado, Palamidessi and Smith [JPS21], present a comprehensive framework to analyze the leakage of deterministic and order-revealing encryption, respectively. Their approach is based on quantitative information flow (QIF) which is a theoretical framework originally proposed to study the information that a program reveals about a secret [Den82, Gra92]. For an introduction to QIF we refer the reader to [ACM+20]. At a very high level, the framework models a leakage pattern as a channel which, together with a prior distribution over the plaintexts, results in a distribution over posterior distributions which the authors call the *hyper-distribution*. This hyper-distribution is known to the adversary and, given some observed leakage, results in a specific posterior distribution. The framework also models different adversarial goals as *gain functions* g which can be thought of as loss functions from decision theory and machine learning. The prior g-vulnerability is then defined as the expected gain with respect to the prior distribution and the posterior g-vulnerability is defined as the expected gain over the hyper-distribution. The g-leakage is then defined as the difference or the quotient of the prior and posterior g-vulnerabilities. The authors study the g-leakage of DTE and ORE for various gain functions and prior distributions and use their Theorems to design and study mitigation techniques. Some results are quite surprising; e.g., the authors are able to show that ideal ORE is safe to use against an adversary that wishes to recover an entire column if the values in the column are sampled uniformly at random and the value space is larger than the number of rows.

Leakage inversion. Closer to our own work and appearing roughly concurrently, Kornaropoulos, Moyer, Papamanthou and Psomas [KMPP22] propose a framework to study the leakage of searchable encryption schemes. Roughly speaking, their approach is to characterize the set of all databases (technically multi-maps) that lead to the same observed leakage as the target with respect to a certain leakage profile. This set is the target database's *reconstruction space* and the logarithm of its size is reported as the amount of information revealed about the target database. The framework of [KMPP22] quantifies leakage with respect to full data recovery attacks against (scheme specific) response identity leakage, which reveals the results of a query. [3] Furthermore, it handles auxiliary information that can be modeled as a predicate and that can be used to filter out items from the reconstruction space (e.g., "the data contains the word crypto").

PAC learning. Grubbs, Lacharite, Minaud and Paterson propose in [GLMP18] to use PAC learning [Val84] as a framework to study approximate data reconstruction attacks.

[3] The response identity is sometimes referred to as the access pattern in the context of searchable symmetric encryption

More precisely, they show how, given $O\left(\frac{d}{\varepsilon}\log\frac{d}{\varepsilon\delta}\right)$ known queries sampled i.i.d, an adversary can recover an ε-approximation of a column with probability at least $1-\delta$. Here, d is the VC-dimension of a concept class needed for the reduction to PAC learning and an ε-approximation is, roughly speaking, a column whose entries will be incorrect with probability at most ε. Similarly to the leakage inversion framework, this approach focuses on data recovery attacks from response identity leakage but, unlike leakage inversion, it only applies to *known-query* attacks.

ϵ-similarity. Another work that closely related to ours is by Damie, Leger, Hahn, and Peter who also describe a (frequentist) statistical framework to analyze leakage in encrypted search [DLHP23]. Their framework, however, differs from ours in several important ways. First, their approach specifically addresses co-occurrence leakage and attacks that rely on the similarity of co-occurrences between secret and auxiliary data. In contrast, our framework and theorems apply to any leakage profile and auxiliary distribution—although our leakage engine is restricted to a particular class of leakages. Second, while their framework focuses on analyzing how an adversary's ability to estimate the secrets' co-occurrence distribution from its auxiliary data impacts the attack, our framework assumes the adversary already possesses an auxiliary *distribution* that could be generated from an auxiliary dataset or not. This auxiliary distribution could represent any arbitrary knowledge the adversary has about the secrets. Third, the main result in [DLHP23] is statistically asymptotic—meaning that it holds for sufficiently large samples—and on the assumption that the auxiliary dataset is sampled from a distribution with the same co-occurrence probabilities as the secret dataset. In addition to their leakage analysis framework, [DLHP23] also proposes a risk assessment framework and alternative methods for empirically evaluating attacks.

q-Leakage analysis. Finally, we mention a recent paper by Boldyreva, Gui and Warinschi who propose a framework to study leakgae in encrypted search [BGW24]. Their approach is inspired by QIF but extends to SSE and STE schemes. Like ours, this framework studies leakage but it does so from a very different perspective and with different goals in mind. Roughly speaking, the q-Leakage framework quantifies the cost of leakage on the security of a construction. This is done by defining quantitative leakage functions (or ql-functions) which award value to the adversary's output. Different ql-functions capture different adversarial goals and the framework then studies the expected value an adversary obtains. Roughly speaking, the ql-functions of [BGW24] correspond to loss functions in our framework but our work is motivated by the analysis of leakage profiles as their own objects of study for the purposes of better understanding the characteristics that make them exploitable.

Summary. With respect to attacks, the BCG/LDG [WP17] and QIF [JS19, JPS21] frameworks model full data recovery attacks against frequency and order leakage. The leakage inversion [KMPP22] and PAC-based frameworks [GLMP18] model full data recovery attacks against response identity leakage while the ϵ-similarity framework [DLHP23] models full query recovery against co-occurrence leakage. In this work, we focus on full and functional query recovery attacks but our framework naturally and easily handles data recovery attacks as well. With respect to auxiliary information, the BCG/LDG framework handles auxiliary distributions that are within a certain statistical distance from the data distribution. The QIF framework assumes the adversary's auxiliary distribution is the same as the data distribution. Leakage inversion studies auxiliary distributions that can be modeled as predicates and the PAC-based framework assumes (non-distributional/perfect) auxiliary knowledge of client queries. The ϵ-similarity framework assumes the adversary

receives an auxiliary dataset sampled from a distribution that is similar to the secret data. Our framework makes no assumption about auxiliary information.

3 Preliminaries

Notation. The set of all binary strings of length n is denoted as $\{0,1\}^n$, and the set of all finite binary strings as $\{0,1\}^*$. We write $x \leftarrow \Sigma$ to represent an element x being sampled from a distribution Σ, and $x \xleftarrow{\$} X$ to represent an element x being sampled uniformly at random from a set X. The output x of an algorithm $\mathcal{A}$ is denoted by $x \leftarrow \mathcal{A}$. Given a sequence $\mathbf{v}$ of n elements, we refer to its ith element as v_i or $\mathbf{v}[i]$. If S is a set then $\#S$ refers to its cardinality and 2^S to its powerset. We denote the set of all functions from domain $\mathbb{X}$ to co-domain $\mathbb{Y}$ by $\mathsf{Func}(\mathbb{X},\mathbb{Y})$ and the set of bijections from $\mathbb{X}$ and $\mathbb{Y}$ by $\mathsf{Bij}(\mathbb{X},\mathbb{Y})$. Given a function $f : \mathbb{X} \to \mathbb{Y}$ and a sequence $\boldsymbol{x} \in \mathbb{X}^n$, we sometimes write $f(\boldsymbol{x})$ to denote the sequence $(f(x_1),\ldots,f(x_n))$. We write $a \stackrel{\circ}{=} b$ to denote that a is defined as b. We denote the falling factorial by $(m)_i$. The identity function over a domain $\mathbb{X}$ is denoted $\mathsf{id}_{\mathbb{X}}$.

Probabilities and random variables. Given a discrete random variable X defined over a probability space $(\Omega,\mathcal{F},\mu)$, we denote its distribution by $p_X(x)$ or $p(x)$ when X is clear. We denote some special distributions with their own symbols, e.g., we write $u_{\mathbb{X}}$ to denote the uniform distribution over a set $\mathbb{X}$. Given two discrete random variables $X : \Omega \to \mathbb{X}$ and $Y : \Omega \to \mathbb{Y}$, we denote the joint distribution of X and Y by $p_{X,Y}(x,y)$ or $p(x,y)$ when X and Y are clear and the distribution of X conditioned on $Y = y$, for some $y \in \mathbb{Y}$, by $p_{X,Y}(x \mid y)$ or $p(x \mid y)$ when X and Y are clear. Given n independent and identically p-distributed random variables $X_1,\ldots,X_n : \Omega \to \mathbb{X}_0$, we denote their joint distribution $p^{\otimes n}$ and refer to the co-domain of the random sequence $\mathbf{X} = (X_1,\ldots,X_n)$ as $\mathbb{X} = \mathbb{X}_1 \times \cdots \times \mathbb{X}_n = \mathbb{X}_0 \times \cdots \times \mathbb{X}_0$. Throughout, we will make use of logarithmic ratios of probabilities of the form $\log(p(x)/q(x))$, where p and q are probability distributions. As is standard in statistics we define $\log(0)$ as $-\infty$. We will denote multivariate random variables $\mathbf{X} = (X_1,\ldots,X_n)$ using bold font and denote their underlying space as $\mathbb{X} = \mathbb{X}_1 \times \cdots \times \mathbb{X}_n$.

Zipf distributions. A random variable X is Zipf distributed with parameter γ if for all $\lambda \in [m]$,

$$\Pr[X = \lambda] = \frac{\lambda^{-\gamma}}{H_{m,\gamma}},$$

where $H_{m,\gamma} = \sum_{i=1}^{m} 1/\lambda^{\gamma}$ is the general form of the harmonic number. When considering Zipf distributions, we usually assume the existence of a permutation $\pi : \mathbb{X} \to [m]$ that maps every element of $\mathbb{X}$ to a rank in $[m]$. We denote by $\mathcal{Z}_{m,\gamma}$ the Zipf distribution over a space $\mathbb{X}$ of size m and parameter γ.

Kullback-Leibler divergence. Given two probability distributions p and q over the same space, their Kullback-Leibler (KL) divergence is defined as

$$\mathsf{KL}(p\|q) = \sum_{x\in\mathbb{X}} p(x)\cdot\log\left(\frac{p(x)}{q(x)}\right),$$

where $\mathsf{KL}(p\|q)$ is defined as 0 when $p(x) = q(x) = 0$ and as $+\infty$ when $p(x) > 0$ and $q(x) = 0$.

A useful divergence. In addition to the KL divergence, we introduce the following measure which will be useful for our purposes. Given a probability distribution p over $\mathbb{X}$ and two subsets $\mathbb{X}_1, \mathbb{X}_2 \subseteq \mathbb{X}$,

$$D\left(p \mathbin{/} \mathbb{X}_1, \mathbb{X}_2\right) \stackrel{\circ}{=} \log\left(\frac{\max_{\boldsymbol{x}\in\mathbb{X}_1} p(x)}{\min_{\boldsymbol{x}\in\mathbb{X}_2} p(x))}\right),$$

where $D(p \mathbin{/} \mathbb{X}_1, \mathbb{X}_2)$ is defined as 0 when $\max\limits_{\boldsymbol{x}\in\mathbb{X}_1} p(x) = \min\limits_{\boldsymbol{x}\in\mathbb{X}_2} p(x) = 0$ and as $+\infty$ when $\max\limits_{\boldsymbol{x}\in\mathbb{X}_1} p(x) > 0$ and $\min\limits_{\boldsymbol{x}\in\mathbb{X}_2} p(x) = 0$. Intuitively, this divergence measures the disparity between the most likely element of $\mathbb{X}_1$ and the least likely element of $\mathbb{X}_2$.

Loss functions. A *loss function* is a function $g : \Theta \times \Theta \to \mathbb{R}$, where Θ is the parameter space, that quantifies the cost of estimating the true parameter $\theta \in \Theta$ as $\hat{\theta}$. In our setting, the loss function will capture the cost that the adversary incurs by inferring a secret $\hat{\mathbf{s}}$ (or a function of $\hat{\mathbf{s}}$) when the real query sequence is $\mathbf{s}$. For instance, we will use the zero-one loss function $\mathsf{zo} : \mathbb{S} \times \mathbb{S} \to \{0, 1\}$ defined as

$$\mathsf{zo}(\mathbf{s}, \hat{\mathbf{s}}) = \begin{cases} 1 & \text{if } \mathbf{s} \neq \hat{\mathbf{s}} \\ 0 & \text{if } \mathbf{s} = \hat{\mathbf{s}} \end{cases}$$

to capture full query recovery.

Statistical estimators. In this work, we will be interested in inferring the *maximum a-posteriori probability* (MAP) estimate which is defined as

$$\mathsf{map}_{\mathbb{S}}(\boldsymbol{\ell}, \alpha) \stackrel{\circ}{=} \arg\max_{\mathbf{s}} \widetilde{p}\left(\mathbf{s} \mid \boldsymbol{\ell}\right) = \arg\max_{\mathbf{s}\in\mathbb{S}} p\left(\boldsymbol{\ell} \mid \mathbf{s}\right) \cdot \alpha(\mathbf{s})$$

where α is the prior distribution which we will refer to as the auxiliary distribution. In the above, we denote by $\widetilde{p}(\mathbf{s} \mid \boldsymbol{\ell})$ the posterior distribution computed using the auxiliary distribution α as opposed to the true secret distribution $p_{\mathbf{S}}$. We will also be interested in the maximum likelihood estimate (MLE) which is defined as

$$\mathsf{mle}_{\mathbb{S}}(\boldsymbol{\ell}) \stackrel{\circ}{=} \arg\max_{\mathbf{s}\in\mathbb{S}} p\left(\boldsymbol{\ell} \mid \mathbf{s}\right)$$

and the MAP test which is defined as

$$\mathsf{tst}_{\mathbb{Y}}(f, \boldsymbol{\ell}, \alpha) \stackrel{\circ}{=} \arg\max_{y\in\mathbb{Y}} \sum_{\mathbf{s}\in f^{-1}(y)} \widetilde{p}\left(\mathbf{s} \mid \boldsymbol{\ell}\right) = \arg\max_{y\in\mathbb{Y}} \sum_{\mathbf{s}\in f^{-1}(y)} p\left(\boldsymbol{\ell} \mid \mathbf{s}\right) \cdot \alpha(\mathbf{s}),$$

where $f : \mathbb{S} \to \mathbb{Y}$ is a recovery function that maps secrets to an output space $\mathbb{Y}$. Note that the MAP estimate outputs an element of the secret space whereas the MAP test outputs an element of the recovery function's co-domain and, specifically, the one with the highest probability according to the posterior distribution. For example, the recovery function f could be the function that outputs the most significant bit of the secret, function that outputs the XOR of the bits of the secret or even a histogram of a multi-variate secret.

Bayesian networks. A *probabilistic model* is a set of random variables together with their joint and marginal distributions. A *probabilistic graphical model* (PGM) is a probabilistic model whose dependencies can be captured and analyzed using a graph with the random variables as vertices and their probabilistic dependencies (e.g., conditional dependence and independence) as edges or lack thereof. The graphical structure provides both a visual and formal way to express complex relationships between the variables in a compact manner

that enables efficient computation and inference. There are two main forms of PGMs: *Markov random fields* and *Bayesian networks* which we now describe.

A Bayesian network $\mathcal{N}_{\mathbf{X}}$ over a multi-variate random variable $\mathbf{X} = (X_1, \ldots, X_n)$ is a directed acyclic graph with the random variables X_i as vertices and directed edges between variables that are conditionally dependent. In addition, each node X_i with incoming edges is labeled with a *conditional probability table* defined as

$$\mathsf{cpt}(X_i) \stackrel{\circ}{=} \Big\{ p(x_i | z_1, \ldots, z_m) \Big\}_{(z_1,\ldots,z_m) \in \mathbb{Z}_1 \times \cdots \times \mathbb{Z}_m},$$

where $Z_1, \ldots, Z_m \in \mathbf{X}$ are the parents of X_i. We can partition the variables $\mathbf{X}$ into a subset of evidence variables $\mathbf{L} \subset \mathbf{X}$, a set of hidden variables $\mathbf{H} \subset \mathbf{X}$ and a set of secret variables $\mathbf{S} \subset \mathbf{X}$ and use the Bayesian network to infer something about the instantiation $\mathbf{s}$ of the secret variables $\mathbf{S}$ given an instantiation $\boldsymbol{\ell}$ of the evidence variables $\mathbf{L}$. In the context of encrytped search, the secret variables will be the query and/or data variables and the evidence variables will be the leakage variables. The power of Bayesian networks comes from the *Bayesian network chain rule* which states that

$$p(\boldsymbol{x}) = \prod_{i=1}^{n} p(x_i \,|\, \mathsf{prnt}(X_i))$$

which is often more efficient to compute than the standard chain rule. In the design of our inference algorithm in Section 8.1, we make use of a notion called *d-separation* [Pea88] which allows one to prove conditional independence between two BN vertices with respect to a set of other vertices. At a high level, two nodes X_1 and X_2 are conditionally independent with respect to a set of nodes $\mathbf{Y}$ if all the undirected paths between X_1 and X_2 are *blocked* by nodes in $\mathbf{Y}$, where nodes in $\mathbf{Y}$ are blocking if they satisfy one of three structural graph properties. In our case, we only need one of these structural properties called the *fork structure* which is satisfied if the nodes in $\mathbf{Y}$ have outgoing edges towards X_1 and X_1. For more about d-separation we refer the reader to [Pea88, KF09].

Leakage patterns and profiles. A leakage profile $\Lambda_\Sigma = (\mathcal{L}_\mathsf{S}, \mathcal{L}_\mathsf{O})$ is composed of a setup leakage $\mathcal{L}_{\text{„}}$ and an operation leakage $\mathcal{L}_\mathsf{O}$. Each of these leakage functions can themselves be functions of various leakage patterns. In this work, all leakage functions and leakage patterns are *stateful*. We recall some leakage patterns that will appear throughout this work and refer the reader to [KMO18] for a more comprehensive treatment:

- the *query equality* takes as input a data structure and a query and reveals if and when the query was repeated;
- the *response length* (or volume) takes as input a data structure and a query and reveals the length of the query's response;

4 Bayesian Leakage Analysis

We model the design of leakage attacks as a statistical inference problem and leakage attacks themselves as statistical estimators.

Basic definitions. A leakage model $\mathbf{X}$ is a multi-variate random variable over a space whose variables can be partitioned into a set of secret variables $\mathbf{S} \subset \mathbf{X}$ over a secret space $\mathbb{S} = \mathbb{S}_1 \times \cdots \times \mathbb{S}_{\#\mathbf{S}}$, a set of hidden or unobservable variables $\mathbf{H} \subset \mathbf{X}$ over a hidden space $\mathbb{H} = \mathbb{H}_1 \times \cdots \times \mathbb{H}_{\#\mathbf{H}}$ and a set of leakage variables $\mathbf{L} \subset \mathbf{X}$ over a leakage space $\mathbb{L} = \mathbb{L}_1 \times \cdots \times \mathbb{L}_{\#\mathbf{L}}$. The probability distribution $p_{\mathbf{S}}(\cdot)$ over $\mathbb{S}$ is the true secret distribution and $\alpha(\cdot)$ is the auxiliary distribution over $\mathbb{S}$.

Full recovery attacks. A *full recovery attack* is a statistical inference algorithm that, given an observed leakage $\boldsymbol{\ell}$ and the auxiliary distribution α, outputs an element of the secret space $\hat{\mathbf{s}}$. The standard MAP and MLE estimators lead to full recovery attacks $\mathcal{A}_{\mathsf{map}}$ and $\mathcal{A}_{\mathsf{mle}}$ that compute and output $\mathsf{map}_{\mathbb{S}}(\boldsymbol{\ell}, \alpha)$ and $\mathsf{mle}_{\mathbb{S}}(\boldsymbol{\ell})$, respectively.

Functional recovery attacks. A *functional recovery attack* is a statistical inference algorithm that, given an observed leakage $\mathbf{L}$, a recovery function f and the auxiliary distribution α, computes f over an element of the secret space. The standard MAP test leads to a functional recovery attack $\mathcal{A}_{\mathsf{tst}}$ that computes and outputs the MAP test $\mathsf{tst}_{\mathbb{Y}}(f, \boldsymbol{\ell}, \alpha)$.

From auxiliary data to auxiliary distributions. In this work, we assume the adversary has access to an auxiliary distribution α. In practice, however, this is not always the case and the adversary could only have a dataset sampled from a distribution that is similar to the secret distribution. We refer to such datasets as *auxiliary datasets* and note that if they are large enough, standard statistical techniques can be used to learn the distribution from the data. This is usually referred to as *empirical Bayes* in the statistics literature. For our purposes, we will therefore assume the adversary has access to an auxiliary distribution, though we note that a more in depth study of how empirical Bayes' affects the adversary's advantage would be interesting.

4.1 Definitions

In this section, we present our statistical leakage analysis framework which includes a notion of adversarial advantage and establishes a bound on the MAP, MLE and MAP test adversaries' advantages.

Inference. To quantify the success of an attack, we first define an experiment that captures the adversary's recovery task as an inference problem. Consider the following probabilistic epxeriment where $\mathbf{X}$ is a leakage model, α is an auxiliary distribution, $f : \mathbb{S} \to \mathbb{Y}$ is a recovery function, $g : \mathbb{S} \times \mathbb{S} \to \mathbb{R}$ is a loss function and $\mathcal{A}$ is an attack:

- $\mathbf{Inf}^{g}_{\mathcal{A},f}(\mathbf{X}, \alpha)$:
  1. the challenger samples $(\mathbf{s}, \boldsymbol{h}, \boldsymbol{\ell}) \leftarrow \mathbf{X}$;
  2. the adversary outputs $\hat{\mathbf{y}} \leftarrow \mathcal{A}(\boldsymbol{\ell}, \alpha)$;
  3. output $g(f(\mathbf{s}), \hat{\mathbf{y}})$;

Refining advantage. The literature on leakage attacks almost exclusively evaluates attacks by reporting the *recovery rate* which, roughly speaking, is defined as the fraction of values from a test set that the attack recovers correctly [IKK12, NKW15, CGPR15, BKM20, GLMP18, GLMP19, LMP17, LMP18]. Reporting the recovery rate, however, is misleading for several reasons. The first and most obvious limitation of the recovery rate is that it cannot be interpreted in isolation but needs to be compared to the probability of guessing the secret. So, instead, one should report the *advantage* of the attack over guessing the secret.

We note, however, that this can still be misleading. The problem is that leakage attacks also leverage auxiliary information so an attack's recovery rate should be compared to guessing the secret *given the auxiliary information and nothing else*. There are several intuitions that support this. First, a cryptographic scheme cannot be expected to hide information about the data and queries that is already revealed by the auxiliary information alone. Second, recall that the purpose of a leakage attack is to extract information *from*

the leakage *given* the auxiliary information. But if one reports the recovery rate alone, then how do we know that the attack extracted the information from the leakage and not from the auxiliary information alone? To make this more concrete, consider a scheme that is zero-leakage, an auxiliary dataset that is exactly the same as the target dataset and an "attack" that outputs the auxiliary dataset. The recovery rate of this attack is 100% but the attack is not even a leakage attack as it is clearly not exploiting leakage.

A concrete example. As an example, we consider the empirical evaluation conducted by Gui, Patterson and Tang in [GPT23]. In this work, the authors describe a leakage attack against a leakage pattern they refer to as the *field-value equality pattern* which, roughly speaking, reveals whether the values of a field are the same or not. [4] To evaluate their attack, the authors choose as target dataset the 2013 American Community Survey (`ACS-13`) and as auxiliary dataset the 2012 American Community Survey (`ACS-12`) both projected on the race (`RAC3P`), state (`ST`), place of birth (`POBP`), place of work (`POWSP`), class of worker (`COW`) and occupation (`OCCP`) fields. We note that [GPT23] does not provide any justification as to why these particular datasets and fields were chosen. In the following discussion, we will define the set $T = \{\texttt{RAC3P}, \texttt{ST}, \texttt{POBP}, \texttt{POWSP}, \texttt{COW}, \texttt{OCCP}\}$ and often refer to the T-projected datasets $\texttt{ACS-13}_T$ and $\texttt{ACS-12}_T$, by which we mean the datasets that result from keeping only the fields in T from `ACS-13` and `ACS-12`, respectively.

Taking a closer look at these datasets, one can see that $\texttt{ACS-13}_T$ and $\texttt{ACS-12}_T$ are very similar under almost any similarity metric one can think of. This can be seen by simple inspection or by using, for example, the multiset Jaccard similarity index which is 0.37 out of a possible 0.5. One can also see that out of $3,113,030$ documents in $\texttt{ACS-12}_T$ and $3,132,795$ documents in $\texttt{ACS-13}_T$, there is a total of $2,333,886$ T-projected *documents* in common. We stress here that this intersection is over the *combination* of all the target fields in T and not over one individual field. In other words, the auxiliary and target databases used in the experiments of [GPT23] share about 75% of their T-projected *documents*.

This raises several questions about how one can even interpret the empirical results of [GPT23]. A first observation is that the only thing one can conclude from them is that the attack is only accurate assuming knowledge of 75% of the target data making it a known-data attack rather than inference attack as claimed. This is similar to the well-known case of the IKK attack which was shown in [CGPR15] to not work as an inference attack as claimed but only as a known-data attack. A more subtle issue, however, is that [GPT23] only report the recovery rate of their attack but not the advantage. The recovery rates are high, ranging from 90% to even 99% but this is very misleading because the trivial attack mentioned above that simply outputs the auxiliary dataset $\texttt{ACS-12}_T$ without even seeing the leakage would achieve somewhere around 75% recovery rate.

Auxiliary guessing. With this in mind, we define a guessing experiment where the secret is estimated as the mode of the auxiliary distribution, which is the best an adversary can do given the auxiliary and assuming it is perfectly accurate in the sense that it is equal to the secret distribution. Consider the following probabilistic experiment where $\mathbf{X}$ is a leakage model, α is an auxiliary distirbution, $f : \mathbb{S} \to \mathbb{Y}$ is a recovery function, $g : \mathbb{Y} \times \mathbb{Y} \to \mathbb{R}$ is a loss function:

- $\mathbf{Guess}^{g}_{f}(\mathbf{X}, \alpha)$:
  1. the challenger samples $(\mathbf{s}, \boldsymbol{h}, \boldsymbol{\ell}) \leftarrow \mathbf{X}$;
  2. computes $\mathcal{G}(\alpha) := \arg\max_{\mathbf{s}' \in \mathbb{S}} \alpha(\mathbf{s}')$;
  3. sets $\hat{\mathbf{s}} \xleftarrow{\$} \mathcal{G}(\alpha)$;

[4] Technically they also study the field-value-contention equality which technically reveals less than the field-value equality but our discussion holds for both leakage patterns.

4. outputs $g\big(f(\mathbf{s}), f(\hat{\mathbf{s}})\big)$;

Auxiliary advantage. We can now formalize our notion of advantage which captures how well a leakage attack recovers the secret compared to guessing. Though our framework can be used with any loss function, our results in this work are for the zero-one loss.

Definition 1. The advantage of an (f, zo)-adversary $\mathcal{A}$ against a leakage model $\mathbf{X}$ with respect to auxiliary distribution α is defined as

$$\mathbf{Adv}^{\mathsf{zo}}_{\mathcal{A},f}(\mathbf{X}, \alpha) \stackrel{\circ}{=} \left|\Pr\left[\,\mathbf{Inf}^{\mathsf{zo}}_{\mathcal{A},f}(\mathbf{X}, \alpha) = 1\,\right] - \Pr\left[\,\mathbf{Guess}^{\mathsf{zo}}_{f}(\mathbf{X}, \alpha) = 1\,\right]\right|.$$

Remark. Notice that the advantage is not an "absolute" notion but a relative one in the sense that it depends on the auxiliary distribution. In other words, an attack could have small advantage against a particular leakage model with respect to a specific auxiliary distribution α but have large advantage with respect to another auxiliary distribution α'.

5 Bounds on Full Recovery Attacks

We prove a Theorem that characterizes the properties of a leakage model that result in $\mathcal{A}_{\mathsf{map}}$ having a small advantage. More precisely, we show that its full-recovery advantage is exponentially bounded by three information-theoretic measures: the *leakage entropy*, the *singular* entropy and the *modal* entropy.

Leakage entropy. The leakage entropy of a leakage model provides a way to quantify the level of indistinguishability between any two secrets with respect to both the leakage they produce and the a-priori beliefs the adversary has in them given the auxiliary distribution.

Definition 2 (Leakage entropy)**.** The leakage entropy of a model $\mathbf{X}$ with respect to an auxiliary distribution α is defined as

$$\Lambda_{\infty}(\mathbf{X}, \alpha) \stackrel{\circ}{=} \min_{\boldsymbol{\ell} \in \mathbb{L},\ \mathbf{s} \in \mathbb{S}_{\boldsymbol{\ell}}} \Lambda_{\boldsymbol{\ell}, \mathbf{s}}(\mathbf{X}, \alpha)$$

where

$$\Lambda_{\boldsymbol{\ell}, \mathbf{s}}(\mathbf{X}, \alpha) \stackrel{\circ}{=} -\log\left(\min_{\mathbf{s}' \in \mathbb{S}_{\boldsymbol{\ell}} \setminus \{\mathbf{s}\}} \left\{ \frac{\mathsf{KL}\Big(p(\boldsymbol{\ell} \mid \mathbf{s}) \,\Big\|\, p(\boldsymbol{\ell} \mid \mathbf{s}')\Big) - \log \alpha(\mathbf{s}')}{-\log \alpha(\mathbf{s})} \right\}\right)$$

where for all $\boldsymbol{\ell} \in \mathbb{L}$, $\mathbb{S}_{\boldsymbol{\ell}} \stackrel{\circ}{=} \{\mathbf{s} \in \mathbb{S} \ :\ p(\boldsymbol{\ell} \mid \mathbf{s}) > 0\}$.

Intuitively, an increase in the leakage entropy reduces the KL divergence between the model's likelihoods which introduces ambiguity in the observed leakage that confuses statistical estimation techniques and hinders the adversary' s ability to infer the secret given its auxiliary information.

Singular entropy. Next, we describe another important aspect of leakage models that affects security. Note that any leakage model that assigns observable leakages to secrets in a one-to-one fashion is immediately information-theoretically exploitable, since, given some observed leakage, one can directly invert it by returning the only secret that could have produced it. Throughout, we will refer to leakages that are in such a one-to-one correspondence with secrets as *singular leakage points*, to their corresponding secrets as *singular secrets* and to the process of recovering a secret from its uniquely associated

leakage as *direct inversion*. An example of direct inversion in the setting of encryption occurs when the message space consists of two messages m_1 and m_2 such that $|m_1| \neq |m_2|$, and one encrypts one of these messages with a length-preserving encryption scheme like the one-time pad. Here, the secrets are the messages and the leakage is the message length and they are in a one-to-one correspondence. In such a scenario, an unbounded adversary can directly invert the encryption—even if it is perfectly secure—by observing the length and returning the associated message. To capture a leakage model's susceptibility to direct inversion, we introduce an information-theoretic measure called the *singular entropy*.

Definition 3 (Singular entropy)**.** The singular entropy of a leakage model $\mathbf{X}$ is defined as

$$\Sigma(\mathbf{X}) \stackrel{\circ}{=} -\log\left(\sum_{\boldsymbol{\ell}\in\mathbb{L}_{\mathbf{1}}} p(\mathbf{s}_{\boldsymbol{\ell}})\right),$$

where $\mathbb{L}_{\mathbf{1}} \stackrel{\circ}{=} \{\boldsymbol{\ell} \in \mathbb{L} \ : \ \#\mathbb{S}_{\boldsymbol{\ell}} = 1\}$ are the model's *singular leakage points*, $\mathbb{S}_{\boldsymbol{\ell}} \stackrel{\circ}{=} \{\mathbf{s} \in \mathbb{S} \ : \ p(\boldsymbol{\ell} \mid \mathbf{s}) > 0\}$ and $\mathbf{s}_{\boldsymbol{\ell}} \in \mathbb{S}$ for $\boldsymbol{\ell} \in \mathbb{L}_{\mathbf{1}}$ is the unique secret such that $p(\boldsymbol{\ell} \mid \mathbf{s}) = p(\mathbf{s} \mid \boldsymbol{\ell}) = 1$.

The singular entropy measures the probability of singular leakage points. Specifically, the singular entropy increases for leakage models that are unlikely to produce singular leakage points. Therefore, higher singular entropy indicates that the model is more secure because it reduces the likelihood of direct inversion. Conversely, the singular entropy decreases for models that are likely to produce singular leakages points. Therefore, lower singular entropy indicates that the model is less secure because it increases the likelihood of direct inversion. In summary, singular entropy is a measure that evaluates leakage model's susceptibility to direct inversion.

Modal entropy. The final definition we need to prove our bound is a notion we call the *modal entropy*. Intuitively, the modal entropy of two distributions p and q measures the belief that p has in the global modes of q.

Definition 4 (Modal entropy)**.** Let $p_{\mathbf{S}}$ and α be two probability distributions over the secret space $\mathbb{S}$. The modal entropy of $p_{\mathbf{S}}$ with respect to α is

$$\chi(p_{\mathbf{S}}, \alpha) = -\log\left(\frac{p_{\mathbf{S}}\left(\mathcal{G}(\alpha)\right)}{\#\mathcal{G}(\alpha)}\right),$$

where $\mathcal{G}(\alpha) \stackrel{\circ}{=} \arg\max_{\mathbf{s}\in\mathbb{S}} \alpha(\mathbf{s})$ are the global modes of α.

We now show some basic but useful identities related to the modal entropy.

Proposition 1. *Let $p_{\mathbf{S}}$ and α be two probability distributions over the secret space $\mathbb{S}$, then*

(i) $\chi(p_{\mathbf{S}}, u_{\mathbb{S}}) = -\log(\#\mathbb{S})$;

(ii) $\chi(p_{\mathbf{S}}, p_{\mathbf{S}}) = H_\infty(\mathbf{S})$.

Proof. Towards showing the first identity, we have

$$\chi(p_{\mathbf{S}}, u) = -\log\left(\frac{p_{\mathbf{S}}\left(\mathcal{G}(u_{\mathbb{S}})\right)}{\#\mathcal{G}(u_{\mathbb{S}})}\right) = -\log\left(\frac{p_{\mathbf{S}}\left(\mathbb{S}\right)}{\#\mathbb{S}}\right) = \log(\#\mathbb{S}).$$

Turning to the second identity, we have

$$\chi(p_{\mathbf{S}}, p_{\mathbf{S}}) = -\log\left(\frac{p_{\mathbf{S}}\left(\mathcal{G}(p_{\mathbf{S}})\right)}{\#\mathcal{G}(p_{\mathbf{S}})}\right) = -\log\left(\frac{\#\mathcal{G}(p_{\mathbf{S}}) \cdot \max_{\mathbf{s}\in\mathbb{S}} p_{\mathbf{S}}(\mathbf{s})}{\#\mathcal{G}(p_{\mathbf{S}})}\right) = \mathrm{H}_\infty(\mathbf{S}).$$

■

Full recovery. We now turn to our Theorem on full recovery which states that the advantage of $\mathcal{A}_{\mathsf{map}}$ against a leakage model is exponentially bounded by the model's leakage, singular and modal entropies. Note that the Theorem does not rely on any independence assumptions.

Theorem 1. *For any leakage model* $\mathbf{X} = (\mathbf{S}, \mathbf{H}, \mathbf{L})$ *and auxiliary distribution* α*,*

$$\mathbf{Adv}^{\mathsf{zo}}_{\mathcal{A}_{\mathsf{map}},\mathsf{id}_{\mathbb{S}}}(\mathbf{X}, \alpha) \leq 2^{-\Lambda_\infty(\mathbf{X},\alpha)} + 2^{-\Sigma(\mathbf{X})} + 2^{-\chi(p_{\mathbf{S}},\alpha)}.$$

Proof. Analyzing the first term of the advantage, we have

$$\begin{aligned}
\Pr\left[\mathbf{Inf}^{\mathsf{zo}}_{\mathcal{A}_{\mathsf{map}},\mathsf{id}_{\mathbb{S}}}(\mathbf{X}, \alpha) = 1\right] &= \Pr\left[\mathcal{A}_{\mathsf{map}}(\mathbf{L}, \alpha) = \mathbf{S}\right] \\
&= \sum_{\mathbf{s}\in\mathbb{S}} \Pr\left[\mathcal{A}_{\mathsf{map}}(\mathbf{L}, \alpha) = \mathbf{s} \,\middle|\, \mathbf{S} = \mathbf{s}\right] \cdot \Pr\left[\mathbf{S} = \mathbf{s}\right] \\
&= \sum_{\boldsymbol{\ell}\in\mathbb{L}} \sum_{\mathbf{s}\in\mathbb{S}} \Pr\left[\mathcal{A}_{\mathsf{map}}(\boldsymbol{\ell}, \alpha) = \mathbf{s} \,\middle|\, \mathbf{S} = \mathbf{s}, \mathbf{L} = \boldsymbol{\ell}\right] \cdot \Pr\left[\mathbf{S} = \mathbf{s}, \mathbf{L} = \boldsymbol{\ell}\right] \\
&= \sum_{\boldsymbol{\ell}\in\mathbb{L}} \sum_{\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} \Pr\left[\mathcal{A}_{\mathsf{map}}(\boldsymbol{\ell}, \alpha) = \mathbf{s} \,\middle|\, \mathbf{S} = \mathbf{s}, \mathbf{L} = \boldsymbol{\ell}\right] \cdot p(\mathbf{s}, \boldsymbol{\ell}) \qquad (1)
\end{aligned}$$

where Equality (1) holds because, for all $\mathbf{s} \in \mathbb{S} \setminus \mathbb{S}_{\boldsymbol{\ell}}$,

$$\Pr\left[\mathcal{A}_{\mathsf{map}}(\mathbf{L}, \alpha) = \mathbf{s} \,\middle|\, \mathbf{S} = \mathbf{s}, \mathbf{L} = \boldsymbol{\ell}\right] = 0$$

since the map adversary can never output a secret $\mathbf{s}$ that does not explain the observed leakage. In the following, we partition the leakage space into its singular leakage points $\mathbb{L}_{\mathbf{1}}$ and its remaining leakage points $\mathbb{L}_{\geq \mathbf{2}}$,

$$\mathbb{L}_{\mathbf{1}} \stackrel{\circ}{=} \{\boldsymbol{\ell} \in \mathbb{L} \;:\; \#\mathbb{S}_{\boldsymbol{\ell}} = 1\} \quad \text{and} \quad \mathbb{L}_{\geq \mathbf{2}} \stackrel{\circ}{=} \{\boldsymbol{\ell} \in \mathbb{L} \;:\; \#\mathbb{S}_{\boldsymbol{\ell}} > 1\}.$$

For clarity of exposition, we will denote the event that $\mathbf{Inf}^{\mathsf{zo}}_{\mathcal{A}_{\mathsf{map}},\mathsf{id}_{\mathbb{S}}}(\mathbf{X}, \alpha)$ outputs 1 as I. We can rewrite Equation (1) as follows,

$$\begin{aligned}
\Pr\left[\mathrm{I}\right] &= \sum_{\boldsymbol{\ell}\in\mathbb{L}_{\mathbf{1}}} p(\mathbf{s}_{\boldsymbol{\ell}}, \boldsymbol{\ell}) + \sum_{\boldsymbol{\ell}\in\mathbb{L}_{\geq \mathbf{2}}} \sum_{\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} \Pr\left[\mathcal{A}_{\mathsf{map}}(\boldsymbol{\ell}, \alpha) = \mathbf{s} \,\middle|\, \mathbf{S} = \mathbf{s}, \mathbf{L} = \boldsymbol{\ell}\right] \cdot p(\mathbf{s}, \boldsymbol{\ell}) \\
&= 2^{-\Sigma(\mathbf{X})} + \sum_{\boldsymbol{\ell}\in\mathbb{L}_{\geq \mathbf{2}}} \sum_{\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} \Pr\left[\mathcal{A}_{\mathsf{map}}(\boldsymbol{\ell}, \alpha) = \mathbf{s} \,\middle|\, \mathbf{S} = \mathbf{s}, \mathbf{L} = \boldsymbol{\ell}\right] \cdot p(\mathbf{s}, \boldsymbol{\ell})
\end{aligned}$$

where the first Equality is because $\#\mathbb{S}_{\boldsymbol{\ell}} = 1$ and $\Pr\left[\mathcal{A}_{\mathsf{map}}(\boldsymbol{\ell}, \alpha) = \mathbf{s} \,\middle|\, \mathbf{S} = \mathbf{s}, \mathbf{L} = \boldsymbol{\ell}\right] = 1$ when $\boldsymbol{\ell} \in \mathbb{L}_{\mathbf{1}}$ (recall that $\mathbf{s}_{\boldsymbol{\ell}}$ is the unique secret that explains the leakage $\boldsymbol{\ell}$) and the second is by definition of the singular entropy.

Now, let $\widetilde{p}(\mathbf{s} \mid \boldsymbol{\ell})$ denote the posterior distribution computed by the adversary, i.e., the MAP estimate computed using the auxiliary distribution α as the prior and let $\mathsf{Sel}_{\mathbf{s}}$ be the event that the secret $\mathbf{s}$ is selected (e.g., uniformly at random) from all the other possible secrets that maximize the posterior probability. We can then write

$$\begin{aligned}
\Pr\left[\mathrm{I}\right] &= 2^{-\Sigma(\mathbf{X})} + \sum_{\boldsymbol{\ell}\in\mathbb{L}_{\geq \mathbf{2}}} \sum_{\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} \Pr_{\boldsymbol{\ell}}\left[\left(\bigwedge_{\mathbf{s}'\neq\mathbf{s}} \widetilde{p}(\mathbf{s} \mid \boldsymbol{\ell}) \geq \widetilde{p}(\mathbf{s}' \mid \boldsymbol{\ell})\right) \bigwedge \mathsf{Sel}_{\mathbf{s}}\right] \cdot p(\mathbf{s}, \boldsymbol{\ell}) \qquad (2) \\
&\leq 2^{-\Sigma(\mathbf{X})} + \sum_{\boldsymbol{\ell}\in\mathbb{L}_{\geq \mathbf{2}}} \sum_{\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} \Pr_{\boldsymbol{\ell}}\left[\bigwedge_{\mathbf{s}'\neq\mathbf{s}} \widetilde{p}(\mathbf{s} \mid \boldsymbol{\ell}) \geq \widetilde{p}(\mathbf{s}' \mid \boldsymbol{\ell})\right] \cdot p(\mathbf{s}, \boldsymbol{\ell})
\end{aligned}$$

$$\leq 2^{-\Sigma(\mathbf{X})} + \sum_{\boldsymbol{\ell}\in\mathbb{L}_{\geq \mathbf{2}}} \sum_{\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} \Pr_{\boldsymbol{\ell}}\left[\bigwedge_{\mathbf{s}'\in\mathbb{S}_{\boldsymbol{\ell}}\setminus\{\mathbf{s}\}} \widetilde{p}(\mathbf{s}\mid\boldsymbol{\ell}) \geq \widetilde{p}(\mathbf{s}'\mid\boldsymbol{\ell})\right]\cdot p(\mathbf{s},\boldsymbol{\ell}) \tag{3}$$

$$\leq 2^{-\Sigma(\mathbf{X})} + \sum_{\boldsymbol{\ell}\in\mathbb{L}_{\geq \mathbf{2}}} \sum_{\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} \Pr_{\boldsymbol{\ell}}\left[\bigwedge_{\mathbf{s}'\in\mathbb{S}_{\boldsymbol{\ell}}\setminus\{\mathbf{s}\}} \log\left(\frac{\widetilde{p}(\mathbf{s}\mid\boldsymbol{\ell})}{\widetilde{p}(\mathbf{s}'\mid\boldsymbol{\ell})}\right) \geq 0\right]\cdot p(\mathbf{s},\boldsymbol{\ell})$$

$$\leq 2^{-\Sigma(\mathbf{X})} + \sum_{\boldsymbol{\ell}\in\mathbb{L}_{\geq \mathbf{2}}} \sum_{\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} \Pr_{\boldsymbol{\ell}}\left[\bigwedge_{\mathbf{s}'\in\mathbb{S}_{\boldsymbol{\ell}}\setminus\{\mathbf{s}\}} \log\left(\frac{p(\boldsymbol{\ell}\mid\mathbf{s})\cdot\alpha(\mathbf{s})}{p(\boldsymbol{\ell}\mid\mathbf{s}')\cdot\alpha(\mathbf{s}')}\right) \geq 0\right]\cdot p(\mathbf{s},\boldsymbol{\ell})$$

$$\leq 2^{-\Sigma(\mathbf{X})} + \sum_{\boldsymbol{\ell}\in\mathbb{L}_{\geq \mathbf{2}}} \sum_{\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} \Pr_{\boldsymbol{\ell}}\left[\bigwedge_{\mathbf{s}'\in\mathbb{S}_{\boldsymbol{\ell}}\setminus\{\mathbf{s}\}} \log\left(\frac{p(\boldsymbol{\ell}\mid\mathbf{s})}{p(\boldsymbol{\ell}\mid\mathbf{s}')}\right) - \log\left(\frac{\alpha(\mathbf{s}')}{\alpha(\mathbf{s})}\right) \geq 0\right]\cdot p(\mathbf{s},\boldsymbol{\ell})$$

$$\leq 2^{-\Sigma(\mathbf{X})} + \sum_{\boldsymbol{\ell}\in\mathbb{L}_{\geq \mathbf{2}}} \sum_{\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} \min_{\mathbf{s}'\in\mathbb{S}_{\boldsymbol{\ell}}\setminus\{\mathbf{s}\}} \Pr_{\boldsymbol{\ell}}\left[\log\left(\frac{p(\boldsymbol{\ell}\mid\mathbf{s})}{p(\boldsymbol{\ell}\mid\mathbf{s}')}\right) - \log\alpha(\mathbf{s}') \geq -\log\alpha(\mathbf{s})\right]\cdot p(\mathbf{s},\boldsymbol{\ell}) \tag{4}$$

where, Inequality 3 follows from the fact that we are only interested in the secrets $\mathbf{s}'$ such that $\widetilde{p}(\mathbf{s}'\mid\boldsymbol{\ell}) > 0$ otherwise $\widetilde{p}(\mathbf{s}\mid\boldsymbol{\ell}) \geq \widetilde{p}(\mathbf{s}'\mid\boldsymbol{\ell}) = 0$ is always true and those secrets do not contribute to the overall probability and Inequality 4 follows from the Fréchet's inequalities. Also, observe that for all $\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}$, when $\#\mathbb{S}_{\boldsymbol{\ell}} > 1$, both $\widetilde{p}(\mathbf{s}\mid\boldsymbol{\ell})$ and $\widetilde{p}(\mathbf{s}'\mid\boldsymbol{\ell})$ are strictly positive which implies that both $\alpha(\mathbf{s})$ and $\alpha(\mathbf{s}')$ are non-negative due to Baye's rule. This is needed to guarantee that the divisions in the inequalities above are valid.

In the following, let

$$\lambda_{\boldsymbol{\ell},\mathbf{s},\mathbf{s}'} \stackrel{\circ}{=} \log\left(\frac{p(\boldsymbol{\ell}\mid\mathbf{s})}{p(\boldsymbol{\ell}\mid\mathbf{s}')}\right) - \log\alpha(\mathbf{s}')$$

Since $\lambda_{\boldsymbol{\ell},\mathbf{s},\mathbf{s}'}$ is a non-negative random variable, using Markov's inequality, we have

$$\Pr_{\boldsymbol{\ell}}\left[\lambda_{\boldsymbol{\ell},\mathbf{s},\mathbf{s}'} \geq -\log\alpha(\mathbf{s})\right] \leq \frac{\mathbb{E}_{\boldsymbol{\ell}}\left[\lambda_{\boldsymbol{\ell},\mathbf{s},\mathbf{s}'}\right]}{-\log\alpha(\mathbf{s})}. \tag{5}$$

Note, however, that

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\ell}}\left[\lambda_{\boldsymbol{\ell},\mathbf{s},\mathbf{s}'}\right] &= \mathop{\mathbb{E}}_{\boldsymbol{\ell}\sim p(\boldsymbol{\ell}\mid\mathbf{s})}\left[\log\left(\frac{p(\boldsymbol{\ell}\mid\mathbf{s})}{p(\boldsymbol{\ell}\mid\mathbf{s}')}\right) - \log\alpha(\mathbf{s}')\right] \\
&= \sum_{\boldsymbol{\ell}} p(\boldsymbol{\ell}\mid\mathbf{s})\cdot\log\left(\frac{p(\boldsymbol{\ell}\mid\mathbf{s})}{p(\boldsymbol{\ell}\mid\mathbf{s}')}\right) - \log\alpha(\mathbf{s}') \\
&= \mathsf{KL}\left(p(\boldsymbol{\ell}\mid\mathbf{s}) \,\middle\|\, p(\boldsymbol{\ell}\mid\mathbf{s}')\right) - \log\alpha(\mathbf{s}').
\end{aligned}$$

Plugging this back into Equation (5), we have

$$\Pr_{\boldsymbol{\ell}}\left[\lambda_{\boldsymbol{\ell},\mathbf{s},\mathbf{s}'} \geq -\log\alpha(\mathbf{s})\right] \leq \left(\mathsf{KL}\left(p(\boldsymbol{\ell}\mid\mathbf{s}) \,\middle\|\, p(\boldsymbol{\ell}\mid\mathbf{s}')\right) - \log\alpha(\mathbf{s}')\right)\cdot(-\log\alpha(\mathbf{s}))^{-1} \tag{6}$$

and plugging Equation (6), in turn, into Equation (4) we have

$$\Pr\left[\,\mathrm{I}\,\right] \leq 2^{-\Sigma(\mathbf{X})} + \sum_{\boldsymbol{\ell}\in\mathbb{L}_{\geq \mathbf{2}}} \sum_{\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} p(\mathbf{s},\boldsymbol{\ell})\cdot 2^{-\Lambda_{\boldsymbol{\ell},\mathbf{s}}(\mathbf{X},\alpha)} \leq 2^{-\Sigma(\mathbf{X})} + 2^{-\Lambda_\infty(\mathbf{X},\alpha)}.$$

where the last inequality follows from the fact that, for all $\ell\in\mathbb{L}$ and $\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}$,

$$\Lambda_{\boldsymbol{\ell},\mathbf{s}}(\mathbf{X},\alpha) \geq \min_{\boldsymbol{\ell}\in\mathbb{L},\ \mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} \Lambda_{\boldsymbol{\ell},\mathbf{s}}(\mathbf{X},\alpha) = \Lambda_\infty(\mathbf{X},\alpha).$$

We now turn to the second term of the advantage. Let $\mathbf{U}_{\mathcal{G}(\alpha)}$ be the random variable that outputs an element of $\mathcal{G}(\alpha) \stackrel{\circ}{=} \arg\max_{\mathbf{s}\in\mathbb{S}} \alpha(\mathbf{s})$ uniformly at random. We then have,

$$\begin{aligned}
\Pr\left[\mathbf{Guess}^{\mathsf{zo}}_{\mathsf{id}_{\mathbb{S}}}(\mathbf{X},\alpha)=1\right] &= \Pr\left[\mathbf{S}=\mathbf{U}_{\mathcal{G}(\alpha)}\right] \\
&= \sum_{\mathbf{s}\in\mathcal{G}(\alpha)} \Pr\left[\mathbf{S}=\mathbf{s} \mid \mathbf{U}_{\mathcal{G}(\alpha)}=\mathbf{s}\right]\cdot u_{\mathcal{G}(\alpha)}(\mathbf{s}) \\
&= \frac{1}{\#\mathcal{G}(\alpha)}\cdot \sum_{\mathbf{s}\in\mathcal{G}(\alpha)} p_{\mathbf{S}}(\mathbf{s}) \\
&= 2^{-\chi(p_{\mathbf{s}},\alpha)}.
\end{aligned}$$

Finally, by the triangle inequality, this gives

$$\left|\Pr\left[\mathbf{Inf}^{\mathsf{zo}}_{\mathcal{A}_{\mathsf{map}},\mathsf{id}_{\mathbb{S}}}(\mathbf{X},\alpha)=1\right]-\Pr\left[\mathbf{Guess}^{\mathsf{zo}}_{\mathsf{id}_{\mathbb{S}}}(\mathbf{X},\alpha)=1\right]\right| \leq 2^{-\Lambda_\infty(\mathbf{X},\alpha)}+2^{-\Sigma(\mathbf{X})}+2^{-\chi(p_{\mathbf{S}},\alpha)},$$

from which the Theorem follows.

■

Bounds for the MLE adversary. We now provide a bound on the advantage of the MLE adversary.

Corollary 1. *For any leakage model* $\mathbf{X}=(\mathbf{S},\mathbf{H},\mathbf{L})$*,*

$$\mathbf{Adv}^{\mathsf{zo}}_{\mathcal{A}_{\mathsf{mle}},\mathsf{id}_{\mathbb{S}}}(\mathbf{X},\perp) \leq 2^{-\Lambda_\infty(\mathbf{X},u_{\mathbb{S}})}+2^{-\Sigma(\mathbf{X})}+\frac{1}{\#\mathbb{S}}$$

where $u_{\mathbb{S}}$ *is the uniform distribution over* $\mathbb{S}$*.*

Proof. By definition of the MLE and MAP we have,

$$\mathsf{mle}_{\mathbb{S}}(\boldsymbol{\ell}) = \arg\max_{\mathbf{s}\in\mathbb{S}} p(\boldsymbol{\ell}\mid\mathbf{s}) = \arg\max_{\mathbf{s}\in\mathbb{S}} p(\boldsymbol{\ell}\mid\mathbf{s})\cdot u_{\mathbb{S}}(\mathbf{s}) = \mathsf{map}_{\mathbb{S}}(\boldsymbol{\ell},u_{\mathbb{S}}).$$

It follows then that

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{zo}}_{\mathcal{A}_{\mathsf{mle}},\mathsf{id}_{\mathbb{S}}}(\mathbf{X},\perp) &= \mathbf{Adv}^{\mathsf{zo}}_{\mathcal{A}_{\mathsf{map}},\mathsf{id}_{\mathbb{S}}}(\mathbf{X},u_{\mathbb{S}}) \\
&\leq 2^{-\Lambda_\infty(\mathbf{X},u_{\mathbb{S}})}+2^{-\Sigma(\mathbf{X})}+2^{-\chi(p_{\mathbf{S}},u_{\mathbb{S}})} \\
&= 2^{-\Lambda_\infty(\mathbf{X},u_{\mathbb{S}})}+2^{-\Sigma(\mathbf{X})}+\frac{1}{\#\mathbb{S}}
\end{aligned}$$

where the inequality follows from Theorem 1 and last equality follows from Proposition 1.

■

Identical auxiliary and secret distributions. In the following corollary, we show a bound for the case when the auxiliary is the same as the secret distribution.

Corollary 2. *For any leakage model* $\mathbf{X}=(\mathbf{S},\mathbf{H},\mathbf{L})$*, if the auxiliary distribution* $\alpha=p_{\mathbf{S}}$ *then*

$$\mathbf{Adv}^{\mathsf{zo}}_{\mathcal{A}_{\mathsf{map}},\mathsf{id}_{\mathbb{S}}}(\mathbf{X},\alpha) \leq 2^{-\Lambda_\infty(\mathbf{X},\alpha)}+2^{-\Sigma(\mathbf{X})}+2^{-H_\infty(\mathbf{S})}$$

Proof. It follows from Theorem 1 and Proposition 1 that,

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{zo}}_{\mathcal{A}_{\mathsf{map}},\mathsf{id}_{\mathbb{S}}}(\mathbf{X},\alpha) &\leq 2^{-\Lambda_\infty(\mathbf{X},p_{\mathbf{S}})}+2^{-\Sigma(\mathbf{X})}+2^{-\chi(p_{\mathbf{S}},p_{\mathbf{S}})} \\
&= 2^{-\Lambda_\infty(\mathbf{X},p_{\mathbf{S}})}+2^{-\Sigma(\mathbf{X})}+2^{-\mathrm{H}_\infty(\mathbf{S})}.
\end{aligned}$$

■

6 Bounds on Functional Recovery Attacks

We now turn to the study of functional recovery attacks. We prove a Theorem that characterizes the properties of a leakage model that lead to $\mathcal{A}_{\mathsf{tst}}$ having small advantage. Similarly to the case of full recovery, we show that the advantage of $\mathcal{A}_{\mathsf{tst}}$ against any leakage model is exponentially bounded by functional variants of the information-theoretic measures defined previously.

Functional leakage entropy. The functional leakage entropy of a leakage model quantifies the difficulty in estimating $f(\mathbf{s})$ given the observed leakage.

Definition 5 (Functional leakage entropy)**.** The functional leakage entropy of a leakage model $\mathbf{X} = (\mathbf{S}, \mathbf{H}, \mathbf{L})$ with respect to a recovery function $f : \mathbb{S} \to \mathbb{Y}$ and an auxiliary distribution α is defined as

$$\Gamma_\infty(\mathbf{X}, f, \alpha) \stackrel{\circ}{=} \min_{\boldsymbol{\ell} \in \mathbb{L},\ \mathbf{s} \in \mathbb{S}_{\boldsymbol{\ell}}} \Gamma_{\mathbf{s}}(\mathbf{X}, f, \alpha)$$

where

$$\Gamma_{\boldsymbol{\ell},\mathbf{s}}(\mathbf{X}, f, \alpha) \stackrel{\circ}{=} -\log \left(\min_{\mathbf{y} \in \mathbb{Y}_{\boldsymbol{\ell}} \setminus \{f(\mathbf{s})\}} \left\{ \frac{\log\left(\frac{\#\mathbb{S}_{\mathbf{s}}}{\#\mathbb{S}_{\mathbf{y}}}\right) + \max_{\boldsymbol{\ell}' \in \mathbb{L}} D\left(p(\boldsymbol{\ell}' \mid \boldsymbol{x}) \,\middle/\, \mathbb{S}_{\mathbf{s}}, \mathbb{S}_{\mathbf{y}}\right)}{-\log \max_{\mathbf{a} \in \mathbb{S}_{\mathbf{s}}} \alpha(\mathbf{a})} \right\} \right),$$

where $\mathbb{S}_{\mathbf{s}} \stackrel{\circ}{=} f^{-1}(f(\mathbf{s}))$, $\mathbb{S}_{\mathbf{y}} \stackrel{\circ}{=} f^{-1}(\mathbf{y})$, $\mathbb{Y}_{\boldsymbol{\ell}} \stackrel{\circ}{=} \{f(\mathbf{s}) \,:\, \mathbf{s} \in \mathbb{S}_{\boldsymbol{\ell}}\}$ and $\mathbb{S}_{\boldsymbol{\ell}} \stackrel{\circ}{=} \{\mathbf{s} \in \mathbb{S} \,:\, p(\mathbf{s} \mid \boldsymbol{\ell}) > 0\}$.

Intuitively, functional leakage entropy characterizes the conditions under which a statistical estimator will return an incorrect output $y \neq f(\mathbf{s})$ when given $\boldsymbol{\ell}$. For ease of exposition, we assume $\#\mathbb{Y}_{\boldsymbol{\ell}} = 1$ so there is only one such y. We call the secrets in $\mathbb{S}_{\mathbf{s}}$ the *valid secrets* since f maps them to $f(\mathbf{s})$ and the secrets in $\mathbb{S}_y$, for $y \neq f(\mathbf{s})$, the *invalid secrets*. Also, we say that leakage $\boldsymbol{\ell}$ is a strong signal for secret $\mathbf{s}$ if the likelihood of $\mathbf{s}$ assigns a large probability to $\boldsymbol{\ell}$; that is, if $p(\boldsymbol{\ell} \mid \mathbf{s})$ is high. Minimizing $\log(\#\mathbb{S}_{\mathbf{s}}/\#\mathbb{S}_y)$ in the expression requires that the set of invalid secrets $\mathbb{S}_y$ be significantly larger than the set of valid secrets. Minimizing the maximum of the divergence term, on the other hand, requires that, in the worst case over the leakages, the probabilities that the likelihoods of the valid secrets assign to the leakage are small and that the probabilities that the likelihoods of the invalid secrets assign to the leakage are large. In other words, that the leakage be a weak signal of the valid secrets and a strong signal of the invalid secrets. Finally, to minimize the denominator, $-\log \max_{\mathbf{a} \in \mathbb{S}_{\mathbf{s}}} \alpha(\mathbf{a})$, the auxiliary distribution must assign small probabilities to the valid secrets.

If these three conditions are met then: (1) we have more invalid secrets than valid ones; (2) the leakage is a stronger signal of the invalid secrets than the valid ones; and (3) the adversary's auxiliary distribution (and therefore its belief) weighs the invalid secrets more than the valid ones.

Functional singular entropy. Functional singular entropy extends the notion of singular entropy by considering not only whether leakages uniquely identify secrets but also whether they uniquely determine some function of the secrets, as defined by a recovery function. Specifically, functional singular entropy measures the extent to which leakage allows an adversary to estimate a function of the secret with certainty, even if the secret itself cannot be directly recovered from the leakage. In the context of functional singular entropy, a leakage point is considered "singular" if all the secrets that can produce that point are mapped to the same value by the recovery function. More precisely, $\boldsymbol{\ell}$ is singular if the

set of recovery values $\mathbb{Y}_{\boldsymbol{\ell}} = \{f(\mathbf{s}) : \mathbf{s} \in \mathbb{S}_{\boldsymbol{\ell}}\}$ is of size one. This means that an unbounded adversary that observes a singular $\boldsymbol{\ell}$ can determine the value of $f(\mathbf{s})$ with certainty, even if multiple secrets are associated with it.

Definition 6 (Functional singular entropy)**.** The functional singular entropy of a leakage model $\mathbf{X}$ is defined as

$$\Phi(\mathbf{X}, f) \stackrel{\circ}{=} -\log\left(\sum_{\boldsymbol{\ell}\in\mathbb{L}_{\mathbf{1},f}}\sum_{\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} p(\mathbf{s},\boldsymbol{\ell})\right),$$

where $\mathbb{L}_{\mathbf{1},f} \stackrel{\circ}{=} \{\ell \in \mathbb{L} \ : \#\mathbb{Y}_{\boldsymbol{\ell}} = 1\}$, $\mathbb{Y}_{\boldsymbol{\ell}} \stackrel{\circ}{=} \{f(\mathbf{s}) \ : \ \mathbf{s} \in \mathbb{S}_{\boldsymbol{\ell}}\}$ and $\mathbb{S}_{\boldsymbol{\ell}} \stackrel{\circ}{=} \{\mathbf{s} \in \mathbb{S} \ : \ p(\mathbf{s} \mid \boldsymbol{\ell}) > 0\}$.

This variant of singular entropy quantifies the probability associated with these functional singular leakage points which reflects the leakage model's susceptibility to direct estimation of the recovery value. A larger functional singular entropy indicates that these singular leakage points are occur with lower probability, which increases security. On the other hand, a smaller functional singular entropy suggests that the model is more vulnerable, because they occur with higher probability.

Functional modal entropy. We now introduce a notion we call the *functional modal entropy* which measures how well the auxiliary distribution's strongest beliefs align with function values under the true distribution. In other words, instead of focusing on individual secrets like the modal entropy, the functional variant focuses on the pre-image of the recovery function.

Definition 7 (Functional modal entropy)**.** Let $p_{\mathbf{S}}$ and α be two probability distributions over the same space $\mathbb{S}$ and $f : \mathbb{S} \to \mathbb{Y}$ be a recovery function. The functional modal entropy of $p_{\mathbf{S}}$ with respect to α is defined as

$$\Psi(f, p_{\mathbf{S}}, \alpha) = -\log\left(\sum_{y\in\mathbb{Y}} p_{\mathbf{S}}(\mathbb{S}_y) \cdot \frac{\#\left(\mathcal{G}(\alpha) \cap \mathbb{S}_y\right)}{\#\mathcal{G}(\alpha)}\right),$$

where $\mathcal{G}(\alpha) \stackrel{\circ}{=} \arg\max_{\mathbf{s}\in\mathbb{S}} \alpha(\mathbf{s})$ are the global modes of α and $\mathbb{S}_y \stackrel{\circ}{=} f^{-1}(y)$.

To better understand the functional modal entropy, consider the case where f is the function that outputs the most significant bit of the secret. The functional modal entropy then measures how well α's strongest beliefs about which secrets have a 0 or 1 as their most significant bit align with how often those bits actually occur according to the true secret distribution $p_{\mathbf{S}}$.

We now show a useful identity related to the functional modal entropy.

Proposition 2. *Let $p_{\mathbf{S}}$ and α be two probability distributions over the secret space $\mathbb{S}$ and $f : \mathbb{S} \to \mathbb{Y}$ be a recovery function, then*

$$\Psi(f, p_{\mathbf{S}}, u_{\mathbb{S}}) = -\log\left(\sum_{y\in\mathbb{Y}} p_{\mathbf{S}}(\mathbb{S}_y) \cdot u_{\mathbb{S}}(\mathbb{S}_y)\right).$$

Proof. We have

$$\Psi(f, p_{\mathbf{S}}, u_{\mathbb{S}}) = -\log\left(\sum_{y\in\mathbb{Y}} p_{\mathbf{S}}(\mathbb{S}_y) \cdot \frac{\#(\mathcal{G}(u_{\mathbb{S}}) \cap \mathbb{S}_y)}{\#\mathcal{G}(u_{\mathbb{S}})}\right)$$

$$= -\log\left(\sum_{y\in\mathbb{Y}} p_{\mathbf{S}}(\mathbb{S}_y)\cdot\frac{\#\mathbb{S}_y}{\#\mathbb{S}}\right)$$
$$= -\log\left(\sum_{y\in\mathbb{Y}} p_{\mathbf{S}}(\mathbb{S}_y)\cdot u_{\mathbb{S}}(\mathbb{S}_y)\right)$$

where the second equality follows from the fact that $\mathcal{G}(u_{\mathbb{S}}) = \mathbb{S}$.

■

Functional recovery. We now prove our Theorem on functional recovery which states that the advantage of $\mathcal{A}_{\mathsf{tst}}$ against a leakage model is exponentially bounded by the model's functional leakage entropy, functional singular entropy and functional modal entropy.

Theorem 2. *For any leakage model* $\mathbf{X}$*, recovery function* $f : \mathbb{S} \to \mathbb{Y}$ *and auxiliary distribution* α*,*

$$\mathbf{Adv}^{\mathsf{zo}}_{\mathcal{A}_{\mathsf{tst}},f}(\mathbf{X},\alpha) \le 2^{-\Gamma_\infty(\mathbf{X},f,\alpha)} + 2^{-\Phi(\mathbf{X},f)} + 2^{-\Psi(f,p_{\mathbf{S}},\alpha)}.$$

Proof. Analyzing the first term of the advantage, we have

$$\begin{aligned}\Pr\left[\,\mathbf{Inf}^{\mathsf{zo}}_{\mathcal{A}_{\mathsf{tst}},f}(\mathbf{X},\alpha)=1\,\right] &= \Pr\left[\,\mathcal{A}_{\mathsf{tst}}(\mathbf{L},\alpha)=f(\mathbf{S})\,\right]\\ &= \sum_{\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} \Pr\left[\,\mathcal{A}_{\mathsf{tst}}(\mathbf{L},\alpha)=f(\mathbf{s})\,\middle|\,\mathbf{S}=\mathbf{s}\,\right]\cdot p(\mathbf{s})\\ &= \sum_{\boldsymbol{\ell}\in\mathbb{L}}\sum_{\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} \Pr\left[\,\mathcal{A}_{\mathsf{tst}}(\boldsymbol{\ell},\alpha)=f(\mathbf{s})\,\middle|\,\mathbf{S}=\mathbf{s},\mathbf{L}=\boldsymbol{\ell}\,\right]\cdot\Pr\left[\,\mathbf{S}=\mathbf{s},\mathbf{L}=\boldsymbol{\ell}\,\right]\end{aligned} \tag{7}$$

We partition the label space into two: $\mathbb{L}_{\mathbf{1}}$ and $\mathbb{L}_{\geq\mathbf{2}}$ defined as follows,

$$\mathbb{L}_{\mathbf{1}} \stackrel{\circ}{=} \{\ell\in\mathbb{L} \,:\, \#\mathbb{Y}_{\boldsymbol{\ell}}=1\} \quad\text{and}\quad \mathbb{L}_{\geq\mathbf{2}} \stackrel{\circ}{=} \{\ell\in\mathbb{L} \,:\, \#\mathbb{Y}_{\boldsymbol{\ell}}>1\},$$

where $\mathbb{Y}_{\boldsymbol{\ell}} \stackrel{\circ}{=} \{f(\mathbf{s}) \;:\; \mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}\}$, $\mathbb{S}_{\boldsymbol{\ell}} \stackrel{\circ}{=} \{\mathbf{s}\in\mathbb{S} \;:\; p(\mathbf{s}\mid\boldsymbol{\ell})>0\}$. Intuitively, $\mathbb{L}_{\mathbf{1}}$ is the set of leakages $\boldsymbol{\ell}$ such that all secrets in $\mathbb{S}_{\boldsymbol{\ell}}$ are mapped to the same recovery value by f. $\mathbb{L}_{\geq\mathbf{2}}$ is the complement. For clarity, we denote the event that $\mathbf{Inf}^{\mathsf{zo}}_{\mathcal{A}_{\mathsf{tst}},f}(\mathbf{X},\alpha)$ outputs 1 by I.

$$\begin{aligned}\Pr[\,\mathrm{I}\,] &= \sum_{\boldsymbol{\ell}\in\mathbb{L}_{\mathbf{1}}}\sum_{\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} p(\mathbf{s},\boldsymbol{\ell}) + \sum_{\boldsymbol{\ell}\in\mathbb{L}_{\geq\mathbf{2}}}\sum_{\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} \Pr_{\boldsymbol{\ell}}\left[\bigwedge_{y\neq f(\mathbf{s})} \widetilde{p}\left(f^{-1}(f(\mathbf{s}))\,\middle|\,\boldsymbol{\ell}\right)\geq\widetilde{p}\left(f^{-1}(y)\,\middle|\,\boldsymbol{\ell}\right)\right]\cdot p(\mathbf{s},\boldsymbol{\ell})\\ &= 2^{-\Phi(\mathbf{X},f)} + \sum_{\boldsymbol{\ell}\in\mathbb{L}_{\geq\mathbf{2}}}\sum_{\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} \Pr_{\boldsymbol{\ell}}\left[\bigwedge_{y\neq f(\mathbf{s})} \log\left(\frac{\widetilde{p}\left(f^{-1}\left(f(\mathbf{s})\right)\,\middle|\,\boldsymbol{\ell}\right)}{\widetilde{p}\left(f^{-1}(y)\,\middle|\,\boldsymbol{\ell}\right)}\right)\geq 0 \bigwedge \mathsf{Sel}_{f(\mathbf{s})}\right]\cdot p(\mathbf{s},\boldsymbol{\ell})\\ &\leq 2^{-\Phi(\mathbf{X},f)} + \sum_{\boldsymbol{\ell}\in\mathbb{L}_{\geq\mathbf{2}}}\sum_{\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} \Pr_{\boldsymbol{\ell}}\left[\bigwedge_{y\neq f(\mathbf{s})} \log\left(\frac{\sum_{\mathbf{a}\in f^{-1}(f(\mathbf{s}))} p(\boldsymbol{\ell}\mid\mathbf{a})\cdot\alpha(\mathbf{a})}{\sum_{\mathbf{a}\in f^{-1}(y)} p(\boldsymbol{\ell}\mid\mathbf{a})\cdot\alpha(\mathbf{a})}\right)\geq 0\right]\cdot p(\mathbf{s},\boldsymbol{\ell})\\ &\leq 2^{-\Phi(\mathbf{X},f)} + \sum_{\boldsymbol{\ell}\in\mathbb{L}_{\geq\mathbf{2}}}\sum_{\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} \left(\min_{y\neq f(\mathbf{s})} \Pr_{\boldsymbol{\ell}}\left[\log\left(\frac{\sum_{\mathbf{a}\in f^{-1}(f(\mathbf{s}))} p(\boldsymbol{\ell}\mid\mathbf{a})\cdot\alpha(\mathbf{a})}{\sum_{\mathbf{a}\in f^{-1}(y)} p(\boldsymbol{\ell}\mid\mathbf{a})\cdot\alpha(\mathbf{a})}\right)\geq 0\right]\right)\cdot p(\mathbf{s},\boldsymbol{\ell})\end{aligned}$$

$$\leq 2^{-\Phi(\mathbf{X},f)} + \sum_{\boldsymbol{\ell}\in\mathbb{L}_{\geq \mathbf{2}}} \sum_{\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} \left(\min_{y\neq f(\mathbf{s})} \Pr_{\boldsymbol{\ell}} \left[\log\left(\frac{\sum_{\mathbf{a}\in f^{-1}(f(\mathbf{s}))} p(\boldsymbol{\ell}\mid \mathbf{a})}{\sum_{\mathbf{a}\in f^{-1}(y)} p(\boldsymbol{\ell}\mid \mathbf{a})} \right) - \log \alpha_y^- \geq -\log \alpha^+_{f(\mathbf{s})} \right] \right) \cdot p(\mathbf{s},\boldsymbol{\ell}), \tag{8}$$

where $\alpha_x^- \stackrel{\circ}{=} \min_{\mathbf{a}\in f^{-1}(x)} \alpha(\mathbf{a})$ $\alpha_x^+ \stackrel{\circ}{=} \max_{\mathbf{a}\in f^{-1}(x)} \alpha(\mathbf{a})$ and the second inequality follows by the Fréchet inequalities. $\mathsf{Sel}_{f(\mathbf{s})}$ is the event that captures the probability to output $f(\mathbf{s})$ in case $\#\mathsf{tst}_{\mathbb{Y}}(f,\boldsymbol{\ell},\alpha) > 0$. Also notice that since we restrict the second quantity above to all $\boldsymbol{\ell}\in\mathbb{L}_{\geq \mathbf{2}}$, then, by definition, there always exists a $y \neq f(\mathbf{s})$ which, in turn, implies that all the divisions above are valid. Let

$$\lambda_{\mathbf{s},y} \stackrel{\circ}{=} \log\left(\frac{\sum_{\mathbf{a}\in f^{-1}(f(\mathbf{s}))} p(\boldsymbol{\ell}\mid \mathbf{a})}{\sum_{\mathbf{a}\in f^{-1}(y)} p(\boldsymbol{\ell}\mid \mathbf{a})} \right) - \log \alpha_y^-, \quad \text{and} \quad \gamma_{\mathbf{s},y} \stackrel{\circ}{=} -\log \alpha^+_{f(\mathbf{s})},$$

and note that for all $\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}$ and $y \neq f(\mathbf{s})$,

$$\begin{aligned} \lambda_{\mathbf{s},y} &\leq \log\left(\frac{\#\mathbb{S}_{\mathbf{s}} \cdot \max_{\mathbf{a}\in f^{-1}(f(\mathbf{s}))} p(\boldsymbol{\ell}\mid \mathbf{a})}{\#\mathbb{S}_y \cdot \min_{\mathbf{a}\in f^{-1}(y)} p(\boldsymbol{\ell}\mid \mathbf{a})} \right) - \log \alpha_y^- && (9)\\ &\leq \log\left(\frac{\#\mathbb{S}_{\mathbf{s}}}{\#\mathbb{S}_y} \right) + \log\left(\frac{\max_{\mathbf{a}\in f^{-1}(f(\mathbf{s}))} p(\boldsymbol{\ell}\mid \mathbf{a})}{\min_{\mathbf{a}\in f^{-1}(y)} p(\boldsymbol{\ell}\mid \mathbf{a})} \right) - \log \alpha_y^-, && (10) \end{aligned}$$

where $\mathbb{S}_{\mathbf{s}} \stackrel{\circ}{=} f^{-1}(f(\mathbf{s}))$ and $\mathbb{S}_y \stackrel{\circ}{=} f^{-1}(y)$.

Given that $\lambda_{\mathbf{s},y}$ is a non-negative random variable and $-\log \alpha^+_{f(\mathbf{s})} > 0$, then by Markov's inequality, we have

$$\begin{aligned} \Pr_{\boldsymbol{\ell}}\left[\lambda_{\mathbf{s},y} > \gamma_{\mathbf{s},y}\right] &\leq \mathop{\mathbb{E}}_{\boldsymbol{\ell}}\left[\lambda_{\mathbf{s},y}\right] \cdot \gamma_{\mathbf{s},y}^{-1} \\ &\leq \left(\log\left(\frac{\#\mathbb{S}_{\mathbf{s}}}{\#\mathbb{S}_y} \right) + \mathop{\mathbb{E}}_{\boldsymbol{\ell}}\left[\log\left(\frac{\max_{\mathbf{a}\in f^{-1}(f(\mathbf{s})} p(\boldsymbol{\ell}\mid \mathbf{a})}{\min_{\mathbf{a}\in f^{-1}(y)} p(\boldsymbol{\ell}\mid \mathbf{a})} \right) \right] - \log \alpha_y^- \right) \cdot \gamma_{\mathbf{s},y}^{-1} \\ &\leq \left(\log\left(\frac{\#\mathbb{S}_{\mathbf{s}}}{\#\mathbb{S}_y} \right) + \max_{\boldsymbol{\ell}}\left\{ \log\left(\frac{\max_{\mathbf{a}\in f^{-1}(f(\mathbf{s})} p(\boldsymbol{\ell}\mid \mathbf{a})}{\min_{\mathbf{a}\in f^{-1}(y)} p(\boldsymbol{\ell}\mid \mathbf{a})} \right) \right\} - \log \alpha_y^- \right) \cdot \gamma_{\mathbf{s},y}^{-1} \end{aligned}$$

Plugging this back into Equation 8, we have

$$\Pr\left[\, \mathbf{Inf}^{\mathsf{zo}}_{\mathcal{A}_{\mathsf{tst}},f}(\mathbf{X}) = 1 \,\right] \leq 2^{-\Phi(\mathbf{X},f)} + \sum_{\boldsymbol{\ell}\in\mathbb{L}_{\geq \mathbf{2}}} \sum_{\mathbf{s}\in\mathbb{S}_{\boldsymbol{\ell}}} p(\mathbf{s},\boldsymbol{\ell}) \cdot 2^{-\Gamma_{\boldsymbol{\ell},\mathbf{s}}(\mathbf{X},f,\alpha)} \leq 2^{-\Phi(\mathbf{X},f)} + 2^{-\Gamma_\infty(\mathbf{X},f,\alpha)}.$$

We now turn to the second term of the advantage. Let $\mathbf{U}_{\mathcal{G}(\alpha)}$ be the random variable that outputs an element of $\mathcal{G}(\alpha) = arg\max_{\mathbf{s}\in\mathbb{S}} \alpha(\mathbf{s})$ uniformly at random. We then have

$$\begin{aligned} \Pr\left[\, \mathbf{Guess}^{\mathsf{zo}}_{\mathbf{A},f}(\mathbf{X},\alpha) = 1 \,\right] &= \Pr\left[\, f(\mathbf{S}) = f\left(\mathbf{U}_{\mathcal{G}(\alpha)}\right) \,\right] \\ &= \sum_{y\in\mathbb{Y}} \Pr\left[f(\mathbf{S}) = y \;\middle|\; f\left(\mathbf{U}_{\mathcal{G}(\alpha)}\right) = y \right] \cdot \Pr\left[f\left(\mathbf{U}_{\mathcal{G}(\alpha)}\right) = y \right] \\ &= \sum_{y\in\mathbb{Y}} \Pr\left[\mathbf{S}\in\mathbb{S}_y \;\middle|\; f\left(\mathbf{U}_{\mathcal{G}(\alpha)}\right) = y \right] \cdot \Pr\left[\mathbf{U}_{\mathcal{G}(\alpha)} \in \mathbb{S}_y \right] \qquad (11) \end{aligned}$$

Note, however that for all $y\in\mathbb{Y}$,

$$\Pr\left[\, \mathbf{U}_{\mathcal{G}(\alpha)} \in \mathbb{S}_y \,\right] = \sum_{\mathbf{s}\in\mathbb{S}_y} \Pr\left[\mathbf{U}_{\mathcal{G}(\alpha)} = \mathbf{s}\right]$$

$$
\begin{aligned}
&= \sum_{\mathbf{s}\in\mathbb{S}_y} \sum_{\mathbf{s}'\in\mathcal{G}(\alpha)} \Pr\left[\mathbf{U}_{\mathcal{G}(\alpha)} = \mathbf{s}' \bigwedge \mathbf{U}_{\mathcal{G}(\alpha)} = \mathbf{s}\right] \\
&= \sum_{\mathbf{s}\in\mathbb{S}_y} \sum_{\mathbf{s}'\in\mathcal{G}(\alpha)} \Pr\left[\mathbf{U}_{\mathcal{G}(\alpha)} = \mathbf{s}' \;\middle|\; \mathbf{U}_{\mathcal{G}(\alpha)} = \mathbf{s}\right] \cdot \Pr\left[\mathbf{U}_{\mathcal{G}(\alpha)} = \mathbf{s}\right] \\
&= \frac{1}{\#\mathcal{G}(\alpha)} \cdot \sum_{\mathbf{s}\in\mathbb{S}_y} \sum_{\mathbf{s}'\in\mathcal{G}(\alpha)} \mathbf{1}(\mathbf{s}=\mathbf{s}') \\
&= \frac{1}{\#\mathcal{G}(\alpha)} \cdot \#\left(\mathcal{G}(\alpha)\cap\mathbb{S}_y\right)
\end{aligned}
$$

Plugging this back into Equation (11) we get

$$
\begin{aligned}
\Pr\left[\,\mathbf{Guess}^{\mathsf{zo}}_{\mathbf{A},f}(\mathbf{X},\alpha)=1\,\right] &= \sum_{y\in\mathbb{Y}} p_{\mathbf{S}}(\mathbb{S}_y)\cdot\frac{\#\left(\mathcal{G}(\alpha)\cap\mathbb{S}_y\right)}{\#\mathcal{G}(\alpha)} \\
&= 2^{-\Psi(f,p_{\mathbf{S}},\alpha)}.
\end{aligned}
$$

and, by the triangle inequality, this gives us

$$
\left|\Pr\left[\,\mathbf{Inf}^{\mathsf{zo}}_{\mathcal{A}_{\mathsf{tst}},\mathbf{A},f}(\mathbf{X})=1\,\right] - \Pr\left[\,\mathbf{Guess}^{\mathsf{zo}}_{\mathbf{A},f}(\mathbf{X})=1\,\right]\right| \leq 2^{-\Gamma_\infty(\mathbf{X},f,\alpha)} + 2^{-\Phi(\mathbf{X},f)} + 2^{-\Psi(f,p_{\mathbf{S}},\alpha)},
$$

from which the Theorem follows.

■

Bounds for the MLE adversary. We now provide a bound on the advantage of the MLE adversary against functional recovery attacks.

Corollary 3. *For any leakage model* $\mathbf{X}=(\mathbf{S},\mathbf{H},\mathbf{L})$ *and recovery function* $f:\mathbb{S}\to\mathbb{Y}$*,*

$$
\mathbf{Adv}^{\mathsf{zo}}_{\mathcal{A}_{\mathsf{mle}},\mathsf{id}_{\mathbb{S}}}(\mathbf{X},\perp) \leq 2^{-\Gamma_\infty(\mathbf{X},f,u_{\mathbb{S}})} + 2^{-\Phi(\mathbf{X})} + 2^{-\Psi(f,p_{\mathbf{S}},u_{\mathbb{S}})}.
$$

This follows from the fact that $\mathsf{mle}_{\mathbb{S}}(\boldsymbol{\ell}) = \mathsf{map}_{\mathbb{S}}(\boldsymbol{\ell},u_{\mathbb{S}})$.

7 Full Recovery Against Query Equality Leakage

In this section, we analyze the full recovery advantage of the MAP adversary against the query equality model. We consider two cases: (1) when the query and auxiliary distributions are Zipf-distributed with the same parameter; and (2) when the query and auxiliary distributions are Zipf-distributed with different parameters. The first case captures settings where the adversary has accurate knowledge of the query distribution and the second settings where it only has some approximate knowledge.

The query equality leakage model. The query equality pattern is the most common leakage pattern revealed by sub-linear ESAs. As such it is the most important patter to study. The query equality can be formalized in many ways but in our model it is a joint distribution $\mathbf{X}_{\mathbf{QEQ}}=(\mathbf{Q},H,\mathbf{L})$, where $\mathbf{S}=\mathbf{Q}=(Q_1,\ldots,Q_n)$ is a random query sequence over a query sequence space $\mathbb{Q}=\mathbb{Q}_1\times\cdots\times\mathbb{Q}_n=\mathbb{Q}_0\times\cdots\times\mathbb{Q}_0$, H is a random bijection from $\mathbb{Q}_0$ to $[\#\mathbb{Q}_0]$ and $\mathbf{L}=(L_1,\ldots,L_n)$ is a random leakage sequence such that, for all $i\in[n]$, $L_i=H(Q_i)$.

Zipf queries and Zipf auxiliairies with the same parameter. We consider the case where $\mathbf{S} = \mathbf{Q}$ is a multivariate random variable composed of n i.i.d. Zipf-distributed queries with parameter γ and α is the product distribution of n Zipf distributions with the same parameter γ.

Theorem 3. *Let $\mathbf{X}_{\mathbf{QEQ}} = (\mathbf{Q}, H, \mathbf{L})$ be a query equality leakage model. For all $n \in \mathbb{N}$ and $m \in \mathbb{N}_{>1}$, if $\mathbf{Q} \sim \mathcal{Z}_{m,\gamma}^{\otimes n}$ and $\alpha = \mathcal{Z}_{m,\gamma}^{\otimes n}$,*

$$\mathbf{Adv}^{\mathsf{zo}}_{\mathcal{A}_{\mathsf{map}},\mathsf{id}_{\mathbb{Q}^n}}(\mathbf{X}_{\mathbf{QEQ}}, \alpha) \leq H_{m,\gamma}^{-n} + \frac{\log H_{m,\gamma}}{\log(m^{\gamma} \cdot H_{m,\gamma})}.$$

Proof. We compute each of the terms from Corollary 2.

Claim. $2^{-\Lambda_{\infty}(\mathbf{X},\alpha)} = \log H_{m,\gamma} / \log(m^{\gamma} \cdot H_{m,\gamma})$.

Based on the definition of leakage entropy, we can write

$$2^{-\Lambda_{\infty}(\mathbf{X},\alpha)} = 2^{-\Lambda_{\boldsymbol{q}_{\mathsf{hi}},\bar{\boldsymbol{\ell}}}(\mathbf{X},\alpha)} = \left(\mathsf{KL}\left(p(\boldsymbol{\ell} \mid \boldsymbol{q}_{\mathsf{hi}}) \,\middle\|\, p(\boldsymbol{\ell} \mid \boldsymbol{q}_{\mathsf{lo}})\right) - \log \alpha(\boldsymbol{q}_{\mathsf{lo}})\right) \cdot (-\log \alpha(\boldsymbol{q}_{\mathsf{hi}}))^{-1},$$

where $\boldsymbol{\ell}^*$, $\boldsymbol{q}_{\mathsf{hi}}$ and $\boldsymbol{q}_{\mathsf{lo}}$ are the leakage and the two query sequences that minimize the leakage entropy where $\boldsymbol{q}_{\mathsf{lo}} \in \mathbb{S}_{\boldsymbol{\ell}^*}$ and $\mathbb{S}_{\boldsymbol{\ell}^*} = \{\boldsymbol{q} \in \mathbb{Q} : p(\boldsymbol{\ell}^* \mid \boldsymbol{q}) > 0\}$. In particular, notice that these values have to minimize both $\mathsf{KL}(p(\boldsymbol{\ell} \mid \boldsymbol{q}) \| p(\boldsymbol{\ell} \mid \boldsymbol{q}'))$ and $-\log \alpha(\boldsymbol{q}')$ and maximize $-\log \alpha(\boldsymbol{q})$. Given that the KL divergence takes values in $\mathbb{R}_{\geq 0}$, we want that $\boldsymbol{q}_{\mathsf{hi}}$ and $\boldsymbol{q}_{\mathsf{lo}}$ lead to a KL divergence equal to zero. In our case, this becomes possible for the following sequences,

$$\boldsymbol{\ell}^* = (i, \ldots, i)\,, \quad \boldsymbol{q}_{\mathsf{hi}} = \left(\pi^{-1}(m), \ldots, \pi^{-1}(m)\right) \quad \text{and} \quad \boldsymbol{q}_{\mathsf{lo}} = \left(\pi^{-1}(1), \ldots, \pi^{-1}(1)\right)$$

for any $i \in [\#\mathbb{Q}_0]$ and where $\pi^{-1}(m)$ is the query with the smallest probability, and $\pi^{-1}(1)$ is the query with the highest probability with respect to the Zipf distribution. Note also that we chose $\boldsymbol{\ell}^*$ such that $\boldsymbol{q}_{\mathsf{hi}}, \boldsymbol{q}_{\mathsf{lo}} \in \mathbb{S}_{\boldsymbol{\ell}^*}$. Moreover, both query sequences lead to one of the m possible leakage sequences in $\{(1, \ldots, 1), (2, \ldots, 2), \ldots, (m, \ldots, m)\}$. More formally, for all $\boldsymbol{\ell} \in \mathbb{L}$, $p(\boldsymbol{\ell} \mid \boldsymbol{q}_{\mathsf{hi}}) = p(\boldsymbol{\ell} \mid \boldsymbol{q}_{\mathsf{lo}}) = 1/m$ and since they are equal, we have

$$\mathsf{KL}\left(p(\boldsymbol{\ell} \mid \boldsymbol{q}_{\mathsf{hi}}) \,\middle\|\, p(\boldsymbol{\ell} \mid \boldsymbol{q}_{\mathsf{lo}})\right) = \sum_{\ell \in \mathbb{L}} p(\boldsymbol{\ell} \mid \boldsymbol{q}_{\mathsf{hi}}) \cdot \log \frac{p(\boldsymbol{\ell} \mid \boldsymbol{q}_{\mathsf{hi}})}{p(\boldsymbol{\ell} \mid \boldsymbol{q}_{\mathsf{lo}})} = 0.$$

Now, since α is Zipf-distributed, $-\log \alpha(\boldsymbol{q}_{\mathsf{lo}})$ reaches its minimum when the query sequence is composed of the most frequent query $\pi^{-1}(1)$. Specifically, we have

$$-\log \alpha(\boldsymbol{q}_{\mathsf{lo}}) = -\log \frac{1}{H_{m,\gamma}^n} = n \cdot \log H_{m,\gamma}.$$

Finally, notice that the quantity $(-\log \alpha(\boldsymbol{q}_{\mathsf{hi}}))^{-1}$ reaches its maximum when the query sequence is composed of the least frequent query $\pi^{-1}(m)$ so we have,

$$-\log \alpha(\boldsymbol{q}_{\mathsf{hi}}) = -\log \left(\frac{m^{-\gamma}}{H_{m,\gamma}}\right)^n = n \cdot \log(m^{\gamma} \cdot H_{m,\gamma}).$$

Claim. $2^{-\Sigma(\mathbf{X})} = \sum_{\boldsymbol{\ell} \in \mathbb{L}_1} p(\boldsymbol{q}_{\boldsymbol{\ell}}) = 0.$

Recall that $\mathbb{L}_1 \stackrel{\circ}{=} \{\boldsymbol{\ell} \in \mathbb{L} \;:\; \#\mathbb{S}_{\boldsymbol{\ell}} = 1\}$ and $\mathbb{S}_{\boldsymbol{\ell}} \stackrel{\circ}{=} \{\boldsymbol{q} \in \mathbb{Q} \;:\; p(\boldsymbol{\ell} \mid \boldsymbol{q}) > 0\}$, for all

$\boldsymbol{\ell} \in \mathbb{L}$ and $n \in \mathbb{N}$. Since $m > 1$, there is no leakage sequence that can result from a unique query sequence. In particular, given a sequence $\boldsymbol{\ell} \in \mathbb{L}$, there are $(m)_\lambda$ possible query sequences that could have potentially led to it, where λ represents the number of unique values in $\boldsymbol{\ell}$. More precisely, since $\lambda \geq 1$ and $m \geq 2$, $(m)_\lambda \geq 2$ which implies that $\mathbb{L}_1 = \emptyset$.

Claim. $2^{-\chi(p_{\mathbf{Q}},\alpha)} = H_{m,\gamma}^{-n}$.

We have by definition,

$$2^{-\chi(p_{\mathbf{Q}},\alpha)} = \frac{p_{\mathbf{Q}}\left(\mathcal{G}(\alpha)\right)}{\#\mathcal{G}(\alpha)} = p_{\mathbf{S}}\left(\boldsymbol{q}_{\mathsf{lo}}\right) = \frac{1}{H_{m,\gamma}^{n}},$$

where

$$\mathcal{G}(\alpha) \stackrel{\circ}{=} \arg\max_{\mathbf{s}\in\mathbb{S}} \alpha(\mathbf{s}) = \{\boldsymbol{q}_{\mathsf{lo}}\},$$

and $\boldsymbol{q}_{\mathsf{lo}} = \left\{\pi^{-1}(1), \ldots, \pi^{-1}(1)\right\}$, and $\pi^{-1}(1)$ is the query with the highest probability with respect to the Zipf distribution. Note that $\mathcal{G}(\alpha)$ is composed of a unique query sequence since the Zipf distribution has the property that its pmf is strictly decreasing.

■

Discussion. The bound of Theorem 3 decreases as the size of the query space m increases for a given γ. To see why, notice that we can approximate the bounds above for all $m \geq 1$ and $\gamma \geq 1$ as follows,

$$\begin{aligned} H_{m,\gamma}^{-n} + \frac{\log H_{m,\gamma}}{\log(m^{\gamma} \cdot H_{m,\gamma})} &\leq H_{m,\gamma}^{-n} + \frac{\log H_{m,1}}{\log(m^{\gamma})} \\ &\leq H_{m,\gamma}^{-n} + \frac{\log(\log m + 1)}{\gamma \cdot \log(m)}, \end{aligned}$$

since $\log m \leq H_{m,1} \leq \log m + 1$ and $H_{m,\gamma} \geq 1$. We also make use of the fact that the generalized Harmonic number decreases when increasing the Zipf parameter, for a fixed m. Also, since the generalized Harmonic number can take values between 1 and $\log m + 1$, the above bound will behave differently depending on the chosen Zipf parameter. If we consider the case where $\gamma = 1$, then the bound is

$$\begin{aligned} H_{m,1}^{-n} + \frac{\log H_{m,1}}{\log(m \cdot H_{m,1})} &\leq (\log m)^{-n} + \frac{\log(\log m + 1)}{\log(m \cdot \log m)} \\ &= O\left(\frac{\log\log m}{\log m}\right), \end{aligned}$$

The same result can be obtained for $\gamma = 2$ as $H_{m,2} = \Theta(\log m)$ [SDT98].

Zipf queries and Zipf auxiliairies with distinct parameters. We now consider the case where the query and auxiliary distributions are Zipf product distributions but with different parameters. The proof of Theorem (1) below is similar to Theorem (3) so we omit the details.

Theorem 1. *Let* $\mathbf{X}_{\mathbf{QEQ}} = (\mathbf{Q}, H, \mathbf{L})$ *be a query equality leakage model. For all* $n \in \mathbb{N}$ *and* $m \in \mathbb{N}_{>1}$*, if* $\mathbf{Q} \sim \mathcal{Z}_{m,\gamma}^{\otimes n}$ *and* $\alpha = \mathcal{Z}_{m,\gamma'}^{\otimes n}$ *where* $\gamma \neq \gamma'$*, then*

$$\mathbf{Adv}^{\mathsf{zo}}_{\mathcal{A}_{\mathsf{map}},\mathsf{id}_{\mathbb{Q}}}(\mathbf{X}_{\mathbf{QEQ}}, \alpha) \leq H_{m,\gamma}^{-n} + \frac{\log H_{m,\gamma'}}{\log(m^{\gamma'} \cdot H_{m,\gamma'})}.$$

Zipf queries and Zipf auxiliairies with distinct parameters and permutations. We now consider the more general case where the query and auxiliary distributions are Zipf product distributions but with different parameters and permutations. In particular, we assume the existence of two permutations $\pi_q : \mathbb{Q} \to [m]$ and $\pi_a : \mathbb{Q} \to [m]$, for both the query and auxiliary distributions, that map every element in $\mathbb{Q}_0$ to a rank in $[m]$. For instance, the rank of a query, as defined by the query distribution, can be different from the rank of a query in the auxiliary distribution.

Theorem 2. *Let $\mathbf{X_{QEQ}} = (\mathbf{Q}, H, \mathbf{L})$ be a query equality leakage model. For all $n \in \mathbb{N}$ and $m \in \mathbb{N}_{>1}$, if $\mathbf{Q} \sim \mathcal{Z}_{m,\gamma}^{\otimes n}$ and $\alpha = \mathcal{Z}_{m,\gamma'}^{\otimes n}$ where $\gamma \neq \gamma'$ and $\pi_q : \mathbb{Q}_0 \to [m]$ and $\pi_a : \mathbb{Q}_0 \to [m]$ the underlying permutations, then*

$$\mathbf{Adv}^{\mathsf{zo}}_{\mathcal{A}_{\mathsf{map}},\mathsf{id}_{\mathbb{Q}}}(\mathbf{X_{QEQ}}, \alpha) \leq \left(\pi_q\left(\pi_a^{-1}(1)\right)^{-\gamma} \cdot H_{m,\gamma}\right)^{-n} + \frac{\log H_{m,\gamma'}}{\log(m^{\gamma'} \cdot H_{m,\gamma'})}.$$

Proof. The proof is the same as the one of Theorem (3) except that we have to define the query sequences that minimize the leakage entropy slightly differently to account for the different permutations. More precisely, we consider

$$\boldsymbol{q}_{\mathsf{hi}} = \left(\pi_a^{-1}(m), \ldots, \pi_q^{-1}(m)\right) \quad \text{and} \quad \boldsymbol{q}_{\mathsf{lo}} = \left(\pi_a^{-1}(1), \ldots, \pi_a^{-1}(1)\right).$$

Given these new sequences, we only need to recompute the value of the modal entropy.

Claim. $2^{-\chi(p_{\mathbf{Q}},\alpha)} = \left(\pi_q\left(\pi_a^{-1}(1)\right)^{-\gamma} \cdot H_{m,\gamma}\right)^{-n}$.

We have by definition,

$$\begin{aligned} 2^{-\chi(p_{\mathbf{Q}},\alpha)} &= \frac{p_{\mathbf{Q}}\left(\mathcal{G}(\alpha)\right)}{\#\mathcal{G}(\alpha)}, \\ &= p_{\mathbf{S}}\left(\boldsymbol{q}_{\mathsf{lo}}\right), \\ &= p_{\mathbf{S}}\left(\left(\pi_a^{-1}(1), \ldots, \pi_a^{-1}(1)\right)\right), \\ &= \left(\frac{\pi_q(\pi_a^{-1}(1))^{-\gamma}}{H_{m,\gamma}}\right)^n. \end{aligned}$$

■

Note that the worst case occurs when the most frequent query in the query distribution is the least frequent query in the query distribution. In this case, $\pi_q\left(\pi_a^{-1}(1)\right)^{-\gamma} = m$.

8 Automated Leakage Attacks

A key feature of our framework is its inherent support for automated analysis which allows the security of leakage profiles to be evaluated algorithmically. In other words, our framework enables us to construct a *leakage attack engine* capable of targeting a wide range of leakage profiles without the need to design custom attacks. Although the engine we describe has limitations, we believe that even introducing the concept and demonstrating the feasibility of such a tool is an important step in improving leakage analysis. We discussed new research directions motivated by our work in Section 1 and mention here some of the impact that advances in leakage attack engines can offer:

- *(complex leakage models)* the theoretical analysis of complex leakage models—such as those based on Theorems 1 and 2 or through more concrete analysis as in Section 7—can be technically challenging. A leakage attack engine can replace or complement this analysis by executing code.

- *(general-purpose attacks)* many leakage profiles in the literature have not been cryptanalyzed due to the high cost and difficulty of designing custom leakage attacks. A leakage attack engine, while not necessarily providing optimal attacks, enables authors to cryptanalyze their own constructions more efficiently.

- *(empirical analysis)* with a leakage attack engine, one can analyze a leakage profile using distributions generated from real-world data and study the auxiliary advantage across a variety of query and/or data and auxiliary distributions.

8.1 A Scalable Inference Algorithm for Hidden Function Networks

In this section, we describe a new inference algorithm that computes MAP estimates for a certain class of leakage models that includes the i.i.d. variants of most of the leakage patterns we are aware of. Though the algorithm is computationally limited, it provides a way to study the security of many real-world leakage profiles against full recovery when the secrets are i.i.d. Recall that, in our framework, leakage profiles and auxiliary information are modeled as probabilistic models. Although our bounds do not rely on specific model assumptions, we can leverage structural features of these models to support the use of efficient algorithms.

Overview. We do this by modeling leakage profiles as Bayesian networks and executing a new inference algorithm we designed to work efficiently on a restricted class of BNs that capture most leakage profiles on to i.i.d. secrets. As described in Section 8.2, a Bayesian network is a probabilistic graphical model with the model's random variables as vertices and directed edges between variables that are conditionally dependent. Bayesian networks are widely used in statistics and machine learning due to their expressiveness and support for efficient algorithms. While they cannot model every conceivable leakage profile, they capture all the profiles we are aware of. In the case that more complex leakage profiles appear in future work, we note that our framework naturally extends to Markov Random Fields (MRF) which are more general than Bayesian networks.

From leakage models to attack networks. As mentioned above, we can construct Bayesian networks $\mathcal{N}_{\mathbf{X}}$ for a large class of leakage models $\mathbf{X} = (\mathbf{S}, \mathbf{H}, \mathbf{L})$ in the standard way: each random variable in $\mathbf{X}$ corresponds to a vertex of $\mathcal{N}_{\mathbf{X}}$ and we include directed edges between variables that are conditionally dependent together with appropriate conditional probability tables. Given a leakage model $\mathbf{X} = (\mathbf{S}, \mathbf{H}, \mathbf{L})$ and an auxiliary distribution α, however, we can also construct the Bayesian network representation of a leakage attack by simply replacing the secret random variables $\mathbf{S}$ with auxiliary random variables $\mathbf{A} \sim \alpha$. In other words, by constructing a standard Bayesian network $\mathcal{N}_{\mathbf{X}}(\alpha) = (\mathbf{A}, \mathbf{H}, \mathbf{L})$ where the vertices are the random variables of $(\mathbf{A}, \mathbf{H}, \mathbf{L})$ and the edges and conditional probability tables are the same as $\mathcal{N}_{\mathbf{X}}$.

Variable elimination. A standard way to compute MAP estimates on BNs is to use the *variable elimination* (VE) algorithm. VE relies on the notion of a *factor*, which is a function that maps the domains of a subset of the random variables to $\mathbb{R}$. Factors are abstractions that generalize both probability distributions and unnormalized distributions. Let $\mathcal{N}_{\mathbf{X}}$ be a BN over a joint random variable $\mathbf{X} = (X_1, \ldots, X_n)$ over $\mathbb{X} = \mathbb{X}_1 \times \cdots, \times \mathbb{X}_n$. If $\boldsymbol{x} = (x_1, \ldots, x_n)$ is an assignment to the variables in $\mathbb{X}$, we write $\boldsymbol{x}_{-i}$ to denote the assignment $\overline{x}$ restricted to all variables in $\mathbb{X}$ except for X_i. The algorithm starts by creating a factor for every random variable, initialized to its conditional probability table (CPT). It then proceeds to iteratively manipulate the factors by transforming them using the following set of operations:

- *restrict*: takes as input a factor $\varphi : \mathbb{V} \to \mathbb{R}$, where $\mathbb{V} \subseteq \mathbb{X}$, and an instantiation a_i of an observed variable X_i and returns a new factor $\varphi' : \mathbb{V} \setminus X_i \to \mathbb{R}$ such that

$$\varphi'(\boldsymbol{x}) = \varphi(\boldsymbol{x} \lhd a_i),$$

  where $\boldsymbol{x} \lhd a_i$ denotes the assignment that results from adding the value a_i to the assignment $\boldsymbol{x}$ in the ith coordinate. The restrict operation conditions the factor on the observed value of X_i which eliminates it from the factor's scope.

- *sum*: takes as input a factor $\varphi : \mathbb{V} \to \mathbb{R}$, where $\mathbb{V} \subseteq \mathbb{X}$, and a variable X_i and returns a new factor $\varphi' : \mathbb{V} \setminus X_i \to \mathbb{R}$ such that

$$\varphi'(\boldsymbol{x}) = \sum_{a_i \in \mathbb{X}_i} \varphi(\boldsymbol{x} \lhd a_i),$$

  The sum operation marginalizes out X_i by summing over all its possible values which removes it from the factor.

- *multiply*: takes as input two factors $\varphi_1 : \mathbb{V}_1 \to \mathbb{R}$ and $\varphi_2 : \mathbb{V}_2 \to \mathbb{R}$, where $\mathbb{V}_1, \mathbb{V}_2 \subseteq \mathbb{X}$, and returns a new factor $\varphi' : \mathbb{V}_1 \cup \mathbb{V}_2 \to \mathbb{R}$ such that

$$\varphi'(\boldsymbol{x}) = \varphi_1(\boldsymbol{x}_{\mathbb{V}_1}) \cdot \varphi_2(\boldsymbol{x}_{\mathbb{V}_2}),$$

  where $\boldsymbol{x}_{\mathbb{V}}$ denotes the assignment that results from removing the value associated with $\mathbb{V}$ from $\boldsymbol{x}$. The multiply operation combines two factors by multiplying their corresponding values which joins their scopes.

- *normalize*: takes as input a factor $\varphi : \mathbb{V} \to \mathbb{R}$, where $\mathbb{V} \subseteq \mathbb{X}$, and returns a new factor $\varphi' : \mathbb{V} \to \mathbb{R}$ where

$$\varphi'(\boldsymbol{x}) = \frac{\varphi(\boldsymbol{x})}{\sum_{\mathbf{a} \in \mathbb{X}} \varphi(\mathbf{a})}.$$

  This multiply operation scales the factor so that the sum of its values is 1 which converts it into a probability distribution.

For our purposes, it suffices to know that the algorithm iteratively applies these operations on a list of factors to compute the posterior distribution of the secret variables given the instantiations of the observed variables. Technically speaking, the VE algorithm computes the posterior distribution of the secret variables as opposed to their MAP estimate but the latter can be recovered from the former by computing an arg max on the posterior.

The algorithm requires $O(v \cdot d^{w+1})$ time and $O(d^{w+1})$ space, where v is the number of variables in $\mathcal{N}_{\mathbf{X}}$, w is the *treewidth* of $\mathcal{N}_{\mathbf{X}}$ and d is the maximum domain size of the variables. In statistics and machine learning, the domain sizes d tend to be small, making the VE algorithm practical for models with low treewidth. In the context of cryptography, however, where the domains of the secret variables are message spaces or query and data spaces in encrypted search, these domains can become very large. Another issue is that, when modeled as Bayesian networks, some leakage patterns can result in variables of exponential size in the query space. This is the case, for example, for the query equality pattern (see Section 7), where the domain of the function variable F is $m!$, where m is the size of the query space. Consequently, the VE algorithm is not practical for most cryptographically-relevant models.

Hidden function networks. To address this, we propose a new simpler and more efficient algorithm for a special class of BNs we refer to as *hidden function networks* (HFN). These networks have the following structure:

- *(i.i.d. secrets)* a set of n i.i.d. secret random variables $\mathbf{S} = (S_1, \ldots, S_n)$;
- *(hidden functions)* a set of hidden random variables $\mathbf{H} = (H_1, \ldots, H_k)$ that sample k functions $h_1, \ldots, h_k$ from function spaces $\mathbb{H}_1, \ldots, \mathbb{H}_k$;
- *(observables)* a set of n observable random variables $\mathbf{L} = (L_1, \ldots, L_n)$ such that, for all $i \in [n]$, $L_i = (h_1(s_i), \ldots, h_k(s_i))$.
- *(edges)* a set of directed edges from the hidden variables to each of the observable variables and, for all $i \in [n]$, an edge from S_i to L_i.

HFNs capture a common class of leakage profiles in the context of i.i.d. queries. In fact, as we will see below, the query equality pattern, the volume pattern and the combination of query equality and volume can all be modeled as HFNs when applied to i.i.d. query sequences.

Our initial experiments using the VE algorithm on the query equality pattern required 47.83 seconds and 98 GB of memory to perform a query recovery attack on a query sequence of length $n = 7$ over a query space of size $m = 7$. Our `Bayle` engine, on the other hand, executes the same experiment in 27.9 ms while using only 0.5 GB of memory. This improvement is not purely algorithmic since our algorithm was implemented in Julia 1.10.5 while the VE algorithm was implemented in Python 3.12, but it serves to illustrate the need to develop a new approach to inference.

Secret MAP via the hidden MAP. As mentioned above, a standard way of computing MAP estimates via the VE algorithm is to use it to compute the posterior distribution and then to return its arg max. For HFNs, however, we observe that if at least one of the hidden functions is a bijection, then the MAP of the secret variables can be computed indirectly by: (1) computing the MAP estimate of the hidden functions; and (2) using the inverses of these functions to find the secrets that map to the observed leakage. Using full query recovery from the query equality pattern as an example, recall that the secret i.i.d. variables are $\mathbf{Q} = (Q_1, \ldots, Q_n)$, the single hidden function H is a uniformly distributed random function over the set of bijections from $\mathbb{Q}_0$ to $[\#\mathbb{Q}_0]$ and $\mathbf{L} = (L_1, \ldots, L_n)$ is the random leakage sequence defined as $L_i = H(Q_i)$, for $i \in [n]$. In this case, our observation is that the MAP estimate $\hat{\boldsymbol{q}} = (\hat{q_1}, \ldots, \hat{q_n})$ of the query sequence given observed leakage $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_n)$ can be computed by first computing the MAP estimate $\hat{h}$ given $\boldsymbol{\ell}$ and then computing $\hat{\mathbf{s}} = \left(\hat{h}^{-1}(\ell_1), \ldots, \hat{h}^{-1}(\ell_n)\right)$. Based on this observation, we can now focus on efficiently computing the MAP estimate of the hidden functions.

Efficiently computing the hidden function MAP. Our goal is to compute

$$\begin{aligned}
\mathsf{map}_{\mathbb{H}}(\boldsymbol{\ell}, p_{\mathbf{H}}) &= \arg\max_{\boldsymbol{h} \in \mathbb{H}} p(\boldsymbol{h} \mid \boldsymbol{\ell}) \\
&= \arg\max_{\boldsymbol{h} \in \mathbb{H}} \frac{p(\boldsymbol{\ell} \mid \boldsymbol{h}) \cdot p(\boldsymbol{h})}{p(\boldsymbol{\ell})} \\
&= \arg\max_{\boldsymbol{h} \in \mathbb{H}} p(\boldsymbol{\ell} \mid \boldsymbol{h}) \cdot p(\boldsymbol{h}) \\
&= \arg\max_{\boldsymbol{h} \in \mathbb{H}} \left\{ p(\boldsymbol{h}) \cdot \prod_{i \in [n]} p(\ell_i \mid \boldsymbol{h}) \right\}
\end{aligned}$$

where

$$\ell_i \stackrel{\circ}{=} (\ell_{i,1}, \ldots, \ell_{i,k}) = (h_1(s_i), \ldots, h_k(s_i))$$

and where the third equality follows from the fact that $p(\boldsymbol{\ell})$ is a constant and the last equality follows from the fact that, for all $i \neq j \in [n]$, L_i is conditionally independent of

L_j given $\mathbf{H}$. This is because the nodes in $\mathbf{H}$ d-separate the variables L_i and L_j based on the fork structure. By definition, we then have

$$\mathsf{map}_{\mathbb{H}}(\boldsymbol{\ell}, p_{\mathbf{H}}) = \arg\max_{\boldsymbol{h}\in\mathbb{H}} \left\{ p(\boldsymbol{h}) \cdot \prod_{i\in[n]} p_S \left(\bigcap_{j=1}^{k} h_j^{-1}(\ell_{i,j}) \right) \right\},$$

where $p_{\mathbf{S}} = p_S^{\otimes n}$, i.e., where p_S is the distribution of a single secret. By using the auxiliary distribution α_S instead of p_S, we can estimate the hidden functions by computing

$$\widetilde{\pi}_{\boldsymbol{h}} \stackrel{\circ}{=} p(\boldsymbol{h}) \cdot \prod_{i\in[n]} \alpha_S \left(\bigcap_{j=1}^{k} h_j^{-1}(\ell_{i,j}) \right) = \prod_{i\in[n]} \sum_{s\in\mathbb{S}_{\boldsymbol{h}(\ell_i)}} \alpha_S(s)$$

for all $\boldsymbol{h} \in \mathbb{H}$, where $\mathbb{S}_{\boldsymbol{h}(\ell_i)} \stackrel{\circ}{=} \bigcap_{j=1}^{k} h_j^{-1}(\ell_{i,j})$, and returning the one with the largest $\widetilde{\pi}_{\boldsymbol{h}}$. Now notice that if some h_j is bijective, then $\#h^{-1}(\ell_{i,j}) = 1$ and, therefore, $\#\mathbb{S}_{\boldsymbol{h}(\ell_i)} = 1$ and $\widetilde{\pi}_{\boldsymbol{h}}$ can be computed even faster. Furthermore, since $\mathbf{A}$ is i.i.d., we can reuse the computations of $\alpha\left(\hat{h}_j^{-1}(\ell_i)\right)$ whenever we encounter another $\ell_j = \ell_i$, for $j \neq i$. The final optimization we make is based on the observation that, given some observed leakage $\boldsymbol{\ell}$, some hidden functions $\boldsymbol{h}$ may be ruled out from the set we maximize over. In other words, instead of computing $\pi_{\boldsymbol{h}}$ for all $\boldsymbol{h} \in \mathbb{F}$, we only need to consider the functions in the support of $p(\boldsymbol{h} \mid \boldsymbol{\ell})$.

Comparison. In cases where at least one hidden function is a bijection, our algorithm requires significantly less computation and memory than VE which, even in the best case, must multiply the CPTs of $H_1, \ldots, H_k$ and A_i for each $i \in [n]$ to determine and restrict the CPT of L_i. This operation alone requires $O(d^{k+1})$ time and space.

As mentioned above, VE requires $O(v \cdot d^{w+1})$ time and $O(d^{w+1})$ space, where v is the number of variables in the network, w is the treewidth of the network and d is the maximum domain size of the variables. For HFNs, the treewidth is dictated by the number of secrets (e.g., the query sequence size) so $w = n$. The number of variables is $v = 2n + k$ and $d = \max_{i\in[k]} \#\mathbb{H}_i$. Using VE to compute a MAP estimate on a generic HFN would therefore require $O((2n+k) \cdot d^{n+1})$ time and $O(d^{n+1})$ space. For the case of query equality, where $k = 1$ and $d = m!$, VE requires $O(n \cdot (m!)^{n+1})$ time and $O((m!)^{n+1})$ space. We note that, in practice, VE implementations can improve their space use for the query equality pattern to $O(m^2 \cdot (m!))$ by storing no more than one factor product between H and each A_i and L_i at a time, but this is still prohibitive. In comparison, our algorithm requires $O(kn \cdot m!)$ time in the worst case and $O(kn + m)$ space. Note that our technique to only consider the functions $\boldsymbol{h} \in \mathbb{H}$ in the support of $p(\boldsymbol{h} \mid \boldsymbol{\ell})$ substantially reduces the time in practice.

8.2 Hidden Function Networks for Common Leakage Patterns

In this section, we describe HFNs that model the common leakage patterns we will proceed to study.

Query equality network. The query equality leakage $\mathbf{X}_{\mathbf{QEQ}} = (\mathbf{Q}, H, \mathbf{L})$, for i.i.d. queries, can be modeled as an HFN $\mathcal{N}^{+}_{\mathbf{QEQ}}$ as follows:

- *(i.i.d. secrets)* a set of n i.i.d. query variables $\mathbf{Q} = (Q_1, \ldots, Q_n)$ over a space $\mathbb{Q}_0^n$;

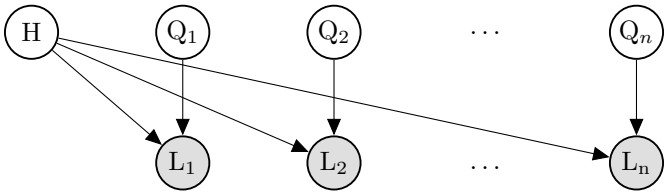

Figure 1: The leakage network $\mathcal{N}^{+}_{\mathbf{QEQ}}$.

- *(hidden functions)* a hidden random variable H that samples a function h uniformly from $\mathsf{Bij}(\mathbb{Q}_0, [\#\mathbb{Q}_0])$;
- *(observables)* a set of n observable random variables $\mathbf{L} = (L_1, \ldots, L_n)$ such that, for all $i \in [n]$, $L_i = h(q_i)$. In other words, each L_i has a conditional probability table of the form
$$p(\ell_i \mid h, q_i) = \begin{cases} 1 & \text{if } \ell_i = h(q_i) \\ 0 & \text{otherwise} \end{cases}$$
- *(edges)* as described in Figure 1, directed edges from H to $(L_1, \ldots, L_n)$ and for all $i \in [n]$, an edge from Q_i to L_i.

Note that, technically, this Bayesian network also reveals the size of the query space $\mathbb{Q}_0$ through the output length of the random bijection h. We add a $+$ in $\mathcal{N}^{+}_{\mathbf{QEQ}}$ to denote this and observe that it is possible to construct a Bayesian network that captures only the query equality.

Volume network. The volume leakage $\mathbf{X}_{\mathbf{VOL}} = (\mathbf{Q}, D, \mathbf{L})$, for i.i.d. queries, can be modeled as an HFN $\mathcal{N}^{+}_{\mathbf{VOL}}$ as follows:

- *(i.i.d. secrets)* a set of n i.i.d. query variables $\mathbf{Q} = (Q_1, \ldots, Q_n)$ over a space $\mathbb{Q}_0^n$;
- *(hidden functions)* a hidden random variable D that samples a function d uniformly from
$$\mathbb{D}_N = \left\{ d \in \mathsf{Func}(\mathbb{Q}_0, [N - m + 1]) : \sum_{q \in \mathbb{Q}_0} d(q) = N \right\}$$
where $m = \#\mathbb{Q}_0$, $N \in \mathbb{N}$ and $N \geq m$.
- *(observables)* a set of n observable random variables $\mathbf{L} = (L_1, \ldots, L_n)$ such that, for all $i \in [n]$, $L_i = d(q_i)$. In other words, each L_i has a conditional probability table of the form
$$p(\ell_i \mid d, q_i) = \begin{cases} 1 & \text{if } \ell_i = d(q_i) \\ 0 & \text{otherwise;} \end{cases}$$
- *(edges)* as described in Figure 2, directed edges from D to $(L_1, \ldots, L_n)$ and for all $i \in [n]$, an edge from Q_i to L_i.

The functions $d \in \mathbb{D}_N$ model the volume of multi-map data structures that map queries (usually called labels) to tuples. N is the size of the multi-map, i.e., the sum of its tuple lengths and $N - m + 1$ is the largest possible tuple size in a N-sized multi-map with m labels. Similar to the query equality, this Bayesian network reveals the size of the query space m as well as the size of the multi-map N through the output length of the functions $d \in \mathbb{D}_N$. We add a $+$ in $\mathcal{N}^{+}_{\mathbf{VOL}}$ to denote this.

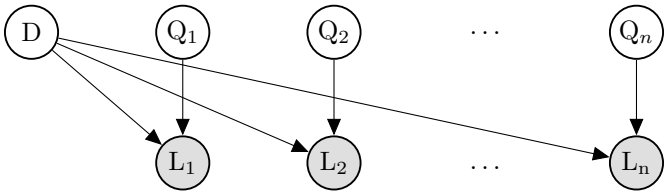

Figure 2: The volume network $\mathcal{N}^{+}_{\mathbf{VOL}}$.

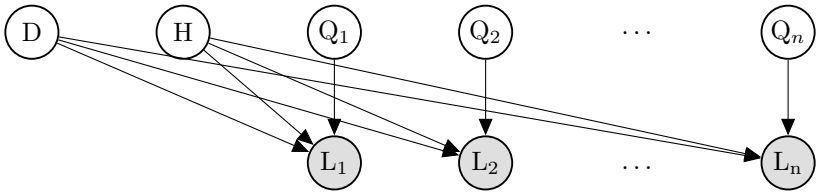

Figure 3: The leakage network $\mathcal{N}^{+}_{\mathbf{QeVo}}$.

Volume and query equality. The combination of volume and query equality $\mathbf{X}_{\mathbf{QeVo}} = (\mathbf{Q}, D, H, \mathbf{L})$ can be modeled as an HFN, $\mathcal{N}^{+}_{\mathbf{QeVo}}$ as follows:

- *(i.i.d. secrets)* a set of n i.i.d. query variables $\mathbf{Q} = (Q_1, \ldots, Q_n)$ over a space $\mathbb{Q}_0^n$;
- *(hidden functions)* a hidden random variable D that samples a function d uniformly from
$$\mathbb{D}_N = \left\{ d \in \mathsf{Func}\left(\mathbb{Q}_0, [N-m+1]\right) : \sum_{q \in \mathbb{Q}_0} d(q) = N \right\}$$
where $m = \#\mathbb{Q}_0$, $N \in \mathbb{N}$ and $N \geq m$ and a hidden random variable H that samples a function h uniformly at random from $\mathsf{Bij}(\mathbb{Q}_0, [\#\mathbb{Q}_0])$.
- *(observables)* a set of n observable random variables $\mathbf{L} = (L_1, \ldots, L_n)$ such that, for all $i \in [n]$, $L_i = \big(d(q_i), h(q_i)\big)$. In other words, each L_i has a conditional probability table of the form
$$p\left(\ell_i \mid d, h, q_i\right) = \begin{cases} 1 & \text{if } \ell_i = \big(h(q_i), d(q_i)\big) \\ 0 & \text{otherwise;} \end{cases}$$
- *(edges)* as described in Figure 3, directed edges from D to $(L_1, \ldots, L_n)$, from H to $(L_1, \ldots, L_n)$ and, for all $i \in [n]$, an edge from Q_i to L_i.

Similar to the query equality and volume cases, we add a $+$ in $\mathcal{N}^{+}_{\mathbf{QeVo}}$ to denote that our network reveals both m and N.

Conjunctions with equality and volume. The leakage of the degree 2 *naive conjunction equality and volume pattern* which reveals the query equality and the volume of the conjuncts can be modeled as an HFN, $\mathcal{N}^{+}_{\mathbf{ConjQeVo}}$, as follows:

- *(i.i.d. secrets)* a set of $2n$ query variables $\mathbf{Q} = (Q_{i,1}, Q_{i,2})_{i \in [n]}$ such that the *pairs* $(Q_{i,1}, Q_{i,2})$ are i.i.d. and where, for all $i[n]$ and $j \in [2]$, $Q_{i,j}$ is over a space $\mathbb{Q}_0$;
- *(hidden functions)* a hidden random variable D that samples a function d uniformly from
$$\mathbb{D}_N = \left\{ d \in \mathsf{Func}\left(\mathbb{Q}_0, [N-m+1]\right) : \sum_{q \in \mathbb{Q}_0} d(q) = N \right\}$$

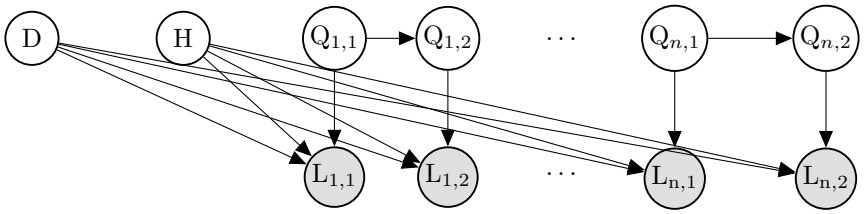

Figure 4: The leakage network $\mathcal{N}^{+}_{\mathbf{ConjQeVo}}$.

where $m = \#\mathbb{Q}_0$, $N \in \mathbb{N}$ and $N \geq m$ and a hidden random variable H that samples a function h uniformly at random from $\mathsf{Bij}(\mathbb{Q}_0, [\#\mathbb{Q}_0])$.

- *(observables)* a set of $2n$ observable random variables $\mathbf{L} = (L_{i,1}, L_{i,2})_{i\in[n]}$ such that, for all $i \in [n]$ and $j \in [2]$, $L_i = (d(q_{i,j}), h(q_{i,j}))$ and such that $Q_{i,1} \neq Q_{i,2}$. In other words, each $L_{i,j}$ has a conditional probability table of the form

$$p\left(\ell_{i,j} \mid d, h, q_{i,j}\right) = \begin{cases} 1 & \text{if } \ell_{i,j} = \big(h(q_{i,j}), d(q_{i,j})\big) \\ 0 & \text{otherwise.} \end{cases}$$

  In addition, each $Q_{i,2}$ has a conditional probability table of the form

$$p\left(q_{i,2} \mid q_{i,1}\right) = \begin{cases} 0 & \text{if } q_{i,1} = q_{i,2} \\ p_{Q_{i,1}}(q_{i,2})/(1 - p_{Q_{i,1}}(q_{i,1})) & \text{otherwise.} \end{cases}$$

- *(edges)* as described in Figure 4, directed edges from D and H to $(L_{i,j})_{i\in[n],j\in[2]}$ and, for all $i \in [n]$, an edge from Q_i to L_i and an edge from $Q_{i,1}$ to $Q_{i,2}$.

Conjunctions with equality and response identity. The degree 2 *naive conjunction equality and response identity* leakage which reveals the query equality and the response identity of the conjuncts can be modeled as an HFN, $\mathcal{N}^{+}_{\mathbf{ConjQeRid}}$, as follows:

- *(i.i.d. secrets)* a set of $2n$ query variables $\mathbf{Q} = (Q_{i,1}, Q_{i,2})_{i\in[n]}$ such that the *pairs* $(Q_{i,1}, Q_{i,2})$ are i.i.d. and where, for all $i \in [n]$ and $j \in [2]$, $Q_{i,j}$ is over a space $\mathbb{Q}_0$;
- *(hidden functions)* a hidden random variable D that samples a function d uniformly from

$$\mathbb{D}_N = \left\{ d \in \mathsf{Func}\left(\mathbb{Q}_0, \bigcup_{k=1}^{N-m+1} \mathbb{I}^k\right) : \sum_{q\in\mathbb{Q}_0} \#d(q) = N \right\}$$

  where $\mathbb{I}$ is a identifier space, $m = \#\mathbb{Q}_0$, $N \in \mathbb{N}$ and $N \geq m$ and a hidden random variable H that samples a function h uniformly at random from $\mathsf{Bij}(\mathbb{Q}_0, [\#\mathbb{Q}_0])$.

- *(observables)* a set of $2n$ observable random variables $\mathbf{L} = (L_{i,1}, L_{i,2})_{i\in[n]}$ such that, for all $i \in [n]$ and $j \in [2]$, $L_i = (d(q_{i,j}), h(q_{i,j}))$ and such that $Q_{i,1} \neq Q_{i,2}$. In other words, each $L_{i,j}$ has a conditional probability table of the form

$$p\left(\ell_{i,j} \mid d, h, q_{i,j}\right) = \begin{cases} 1 & \text{if } \ell_{i,j} = \big(h(q_{i,j}), d(q_{i,j})\big) \\ 0 & \text{otherwise.} \end{cases}$$

  In addition, each $Q_{i,2}$ has a conditional probability table of the form

$$p\left(q_{i,2} \mid q_{i,1}\right) = \begin{cases} 0 & \text{if } q_{i,1} = q_{i,2} \\ p_{Q_{i,1}}(q_{i,2})/(1 - p_{Q_{i,1}}(q_{i,1})) & \text{otherwise.} \end{cases}$$

- *(edges)* as described in Figure 4, directed edges from D and H to $(L_{i,j})_{i\in[n],j\in[2]}$ and, for all $i \in [n]$, an edge from Q_i to L_i and an edge from $Q_{i,1}$ to $Q_{i,2}$.

8.3 Our Attack Engine

We now describe our `Bayle` attack engine which includes two components: (1) a parallel implementation of our HFN inference algorithm; and (2) an experimental harness to empirically evaluate HFNs. `Bayle`[5] is implemented in Julia `1.10.5` and incorporates the following optimizations:

- *SIMD operations:* the values $\pi_{\boldsymbol{h}}$ can be computed in parallel, so we utilize SIMD instructions on a vector that stores, in each coordinate, the products of $\pi_{\boldsymbol{h}}$ computed thus far.
- *floating-point precision:* due to the algorithm's large number of multiplications of probabilities and limited floating-point precision, the computations of the $\pi_{\boldsymbol{h}}$ values tend to approach zero. To address this, we periodically normalize the running values so that they sum to 1. This normalization step cannot be executed in parallel, however, which results in some loss of the benefits provided by our SIMD-based computation. Using trial and error, we determined that 8 multiplications could be handled before normalization was needed.

Experimental harness. To evaluate a leakage model, `Bayle` runs a series of trials, each of which does the following:

1. samples hidden function $\boldsymbol{h} = (h_1, \ldots, h_k)$ and a sequence of queries $\boldsymbol{q} = (q_1, \ldots, q_n)$;
2. computes the observed leakages $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_n)$, where $\ell_i = (h_1(q_i), \ldots, h_k(q_i))$;
3. runs the inference algorithm from Section 8.1;
4. if the inferred sequence $\hat{\boldsymbol{q}}$ is equal to $\boldsymbol{q}$, the the trial succeeds. (We do not consider partial query recovery in our experiments—if any single element of $\hat{\boldsymbol{q}}$ is not equal to $\boldsymbol{q}$, the trial fails.)

Our engine provides an experimental harness that uses Google Cloud Batch to distribute and synchronize a large number of trials for each experiment across up to $2,500$ virtual machines at once. Currently, `Bayle` runs experiments on `e2-standard-8` hosts, each of which has 4 vCPUs and 16 GBs of memory. We explored multi-threading for additional parallelization within individual trials but decided against it for two reasons. First, Google Cloud Batch automatically runs multiple batch tasks on one machine if the machine has sufficient vCPUs and memory and we found that running multiple single-threaded trials on one `e2-standard-8` host was more efficient than running one multi-threaded trial due to the synchronization needed to retrieve the final results of a trial. Second, the normalization step needed to mitigate the limited precision of floating point multiplications would require additional synchronization between threads which would add complexity for limited benefit.

9 Experimental Results

We ran our `Bayle` engine against the $\mathcal{N}^{+}_{\mathbf{QEQ}}$, $\mathcal{N}^{+}_{\mathbf{QeVo}}$, $\mathcal{N}^{+}_{\mathbf{ConjQeVo}}$, and $\mathcal{N}^{+}_{\mathbf{ConjQeRid}}$ networks. Using the experimental harness described in Section 8.3 with a Zipf real query distribution (with $\gamma = 2$) and a uniform real data distribution, we evaluated each network against four

[5]The engine will be made available pending acceptance.

different auxiliary query distributions: a Zipf distribution with $\gamma = 2$ (identical to the real query distribution), a Zipf distribution with $\gamma = 4$ (with the same relative ordering of the domain), a Zipf distribution with $\gamma = 2$ with a different ordering of the domain from that of the real query distribution, and a uniform distribution. For each auxiliary distribution, we varied the size of the query space m from 4 to 10 and the number of queries n from 1000 to 15000. For the $\mathcal{N}_{\mathbf{QeVo}}$, $\mathcal{N}_{\mathbf{ConjQeVo}}$, and $\mathcal{N}_{\mathbf{ConjQeRid}}$ networks, we fix $N = 1000$. For the $\mathcal{N}_{\mathbf{ConjQeVo}}$ and $\mathcal{N}_{\mathbf{ConjQeRid}}$ networks, the real query distribution outputs queries containing exactly two subclauses and is restricted to queries where both subclauses are different.

For each experiment, we ran between 5,000 to 10,000 trials and counted the number of successes. In total, we ran over 8 million trials over 896 experiments. Figure 5 reports the percentage of successes in each experiment.

Discussion. Our experiments demonstrate that the most important factor that determines the MAP adversary's advantage with respect to full query recovery is whether or not the auxiliary query distribution α is the same (or close) to that of the real query distribution. This can be seen by comparing the first two columns of Figure 5 (which use auxiliary query distributions that are the same as or close to the real query distribution) to the last two columns of Figure 5. Interestingly, we see that some auxiliary distributions can cause the success rate to be slightly *worse* than the uniform distribution—the adversary never succeeds in any of the experiments when it uses an Zipf auxiliary query distribution with a different domain permutation from that of the real query distribution. Aside from choosing a "good" auxiliary distribution, the experiments show that the adversary's full recovery rate increases when: (1) the adversary sees longer query sequences (since the adversary sees more leakage), and (2) the size of the query space is smaller (since the number of possible "incorrect" choices decreases with smaller query space). For specific experiments, we highlight the following:

- Given an auxiliary distribution α, the query recovery rates for the query equality pattern $\mathcal{N}_{\mathbf{QEQ}}$ and the query equality and volume pattern $\mathcal{N}_{\mathbf{QeVo}}$ are nearly identical, showing that the additional leakage added by volume for each query does not provide additional help with full query recovery when the data distribution is uniform.

- Given fixed auxiliary distribution α, the recovery rates for both conjunction leakage patterns are nearly the same. This shows that, even though the query equality and response identity pattern $\mathcal{N}_{\mathbf{ConjQeRid}}$ intuitively seems "more leaky" than the query equality and volume pattern $\mathcal{N}_{\mathbf{ConjQeVo}}$, the response identifiers do not assist with query recovery when the real data distribution is uniform.

- For the conjunction networks, we achieve high recovery rate even when the auxiliary distribution does not account for the dependency between sub-clauses in the real query distribution. This shows that, even though our experiemnts do not account for the dependencies between subclause query variables in the auxiliary distribution, the MAP adversary still can achieve high recovery rate when the auxiliary query distribution is close enough to the distribution of the individual subclauses.

References

[ACM+20] Mário S Alvim, Konstantinos Chatzikokolakis, Annabelle McIver, Carroll Morgan, Catuscia Palamidessi, and Geoffrey Smith. *The Science of Quantitative Information Flow*. Springer, 2020.

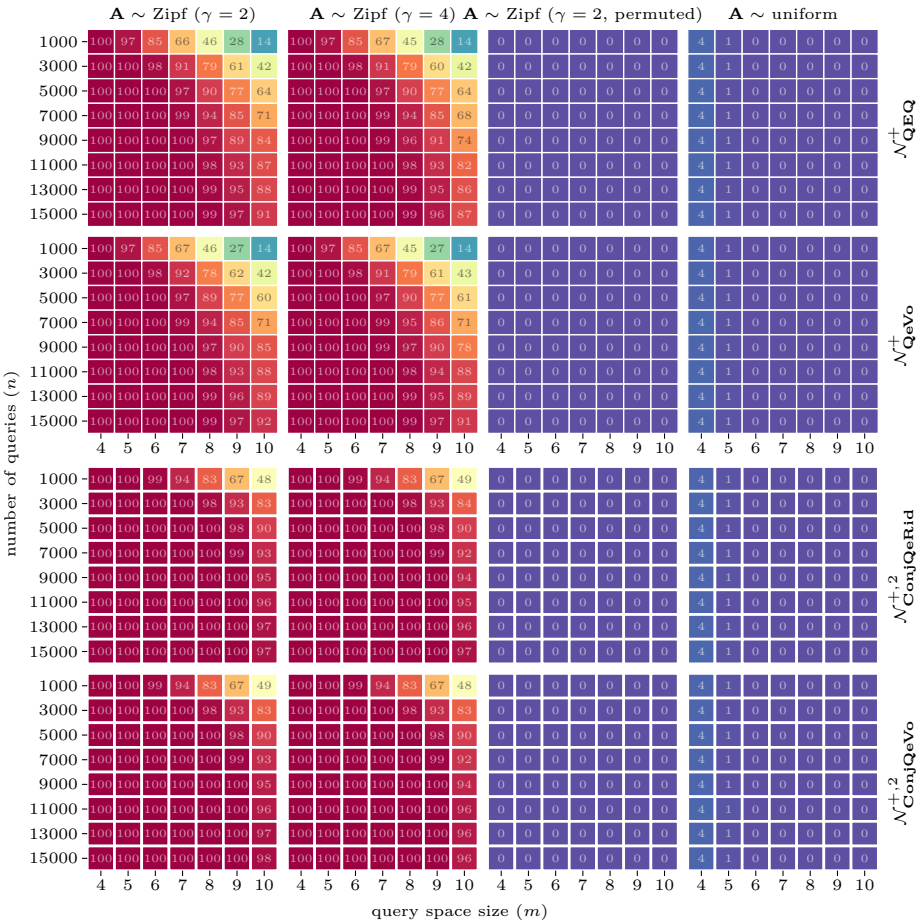

Figure 5: $\mathcal{A}_{\mathsf{map}}$ advantage against different leakage networks.

[APP+21] Ghous Amjad, Sarvar Patel, Giuseppe Persiano, Kevin Yeo, and Moti Yung. Dynamic volume-hiding encrypted multi-maps with applications to searchable encryption. *Cryptology ePrint Archive*, 2021.

[APRZB11] R. Ada Popa, C. Redfield, N. Zeldovich, and H. Balakrishnan. CryptDB: Protecting confidentiality with encrypted query processing. In *ACM Symposium on Operating Systems Principles (SOSP)*, pages 85–100, 2011.

[Bar16] Boaz Barak. Bayesianism, frequentism, and the planted clique, or do algorithms believe in unicorns? Windows on Theory blog, 2016. URL: https://windowsontheory.org/2016/04/13/bayesianism-frequentism-and-the-planted-clique-or-do-algorithms-believe-in-unicorns/.

[BB20] Matthew S. Brennan and Guy Bresler. Reducibility and statistical-computational gaps from secret leakage. In Jacob D. Abernethy and Shivani Agarwal, editors, *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125 of *Proceedings of Machine Learning Research*, pages 648–847. PMLR, 2020. URL: http://proceedings.mlr.press/v125/brennan20a.html.

[BBH18] Matthew S. Brennan, Guy Bresler, and Wasim Huleihel. Reducibility and computational lower bounds for problems with planted sparse structure. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Conference On Learning Theory, COLT 2018, Stockholm, Sweden, 6-9 July 2018*, volume 75 of *Proceedings of Machine Learning Research*, pages 48–166. PMLR, 2018. URL: http://proceedings.mlr.press/v75/brennan18a.html.

[BBH+21] Matthew S. Brennan, Guy Bresler, Samuel B. Hopkins, Jerry Li, and Tselil Schramm. Statistical query algorithms and low degree tests are almost equivalent. In Mikhail Belkin and Samory Kpotufe, editors, *Conference on Learning Theory, COLT 2021, 15-19 August 2021, Boulder, Colorado, USA*, volume 134 of *Proceedings of Machine Learning Research*, page 774. PMLR, 2021. URL: http://proceedings.mlr.press/v134/brennan21a.html.

[BBMM18] Marshall Ball, Elette Boyle, Tal Malkin, and Tal Moran. Exploring the boundaries of topology-hiding computation. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 294–325. Springer, 2018. doi:10.1007/978-3-319-78372-7_10.

[BGI+01] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. In *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2001.

[BGW24] Alexandra Boldyreva, Zichen Gui, and Bogdan Warinschi. Understanding leakage in searchable encryption: a quantitative approach. *Proc. Priv. Enhancing Technol.*, 2024(4):503–524, 2024. URL: https://doi.org/10.56553/popets-2024-0127, doi:10.56553/POPETS-2024-0127.

[BHK+19] Boaz Barak, Samuel Hopkins, Jonathan Kelner, Pravesh K. Kothari, Ankur Moitra, and Aaron Potechin. A nearly tight sum-of-squares lower bound for the planted clique problem. *SIAM Journal on Computing*, 48(2):687–735,

2019. arXiv:https://doi.org/10.1137/17M1138236, doi:10.1137/17M1138236.

[BKM20] Laura Blackstone, Seny Kamara, and Tarik Moataz. Revisiting leakage abuse attacks. In *Network and Distributed System Security Symposium (NDSS '20)*, 2020.

[CGKO06] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky. Searchable symmetric encryption: Improved definitions and efficient constructions. In *ACM Conference on Computer and Communications Security (CCS '06)*, pages 79–88. ACM, 2006.

[CGKS95] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *IEEE Symposium on Foundations of Computer Science (FOCS '95)Symposium on Foundations of Computer Science*, pages 41–. IEEE Computer Society, 1995.

[CGPR15] D. Cash, P. Grubbs, J. Perry, and T. Ristenpart. Leakage-abuse attacks against searchable encryption. In *ACM Conference on Communications and Computer Security (CCS '15)*, pages 668–679. ACM, 2015.

[Cha81] David L Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[CJJ+14] David Cash, Joseph Jaeger, Stanislaw Jarecki, Charanjit Jutla, Hugo Krawczyk, Marcel Rosu, and Michael Steiner. Dynamic searchable encryption in very-large databases: Data structures and implementation. In *Network and Distributed System Security Symposium (NDSS '14)*, 2014.

[CK10] M. Chase and S. Kamara. Structured encryption and controlled disclosure. In *Advances in Cryptology - ASIACRYPT '10*, volume 6477 of *Lecture Notes in Computer Science*, pages 577–594. Springer, 2010.

[Den82] *Cryptography and data security*, volume 112. Addison-Wesley, 1982.

[DLHP23] Marc Damie, Jean-Benoist Leger, Florian Hahn, and Andreas Peter. The statistical nature of leakage in SSE schemes and its role in passive attacks. Cryptology ePrint Archive, Paper 2023/1883, 2023. URL: https://eprint.iacr.org/2023/1883.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference (TCC '06)*, pages 265–284, 2006.

[FIM+06] Joan Feigenbaum, Yuval Ishai, Tal Malkin, Kobbi Nissim, Martin J. Strauss, and Rebecca N. Wright. Secure multiparty computation of approximations. *ACM Trans. Algorithms*, 2(3):435–472, July 2006. doi:10.1145/1159892.1159900.

[FVK+15] B. A Fisch, B. Vo, F. Krell, A. Kumarasubramanian, V. Kolesnikov, T. Malkin, and S. M. Bellovin. Malicious-client security in blind seer: a scalable private dbms. In *IEEE Symposium on Security and Privacy*, pages 395–410. IEEE, 2015.

[GKM21] Marilyn George, Seny Kamra, and Tarik Moataz. Structured encryption and dynamic leakage suppression. In *Advances in Cryptology - EUROCRYPT 2021*, 2021.

[GLMP18] Paul Grubbs, Marie-Sarah Lacharité, Brice Minaud, and Kenneth G. Paterson. Pump up the volume: Practical database reconstruction from volume leakage on range queries. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, pages 315–331. ACM, 2018. doi:10.1145/3243734.3243864.

[GLMP19] Paul Grubbs, Marie-Sarah Lacharité, Brice Minaud, and Kenneth G. Paterson. Learning to reconstruct: Statistical learning theory and encrypted database attacks. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 1067–1083. IEEE, 2019. doi:10.1109/SP.2019.00030.

[GM82] S. Goldwasser and S. Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *Proceedings of the ACM Symposium on Theory of Computing (STOC 1982)*, pages 365–377. ACM Press, 1982.

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play ANY mental game. In *ACM Symposium on the Theory of Computation (STOC '87)*, pages 218–229. ACM, 1987. doi:http://doi.acm.org/10.1145/28395.28420.

[GO96] O. Goldreich and R. Ostrovsky. Software protection and simulation on oblivious RAMs. *Journal of the ACM*, 43(3):431–473, 1996.

[GPT23] Zichen Gui, Kenneth Paterson, and Tianxin Tang. Security analysis of mongodb queryable encryption, 2023.

[Gra92] James W. Gray. Toward a mathematical foundation for information flow security. *J. Comput. Secur.*, 1(3–4):255–294, may 1992.

[GRR19] Adam Groce, Peter Rindal, and Mike Rosulek. Cheaper private set intersection via differentially private leakage. *Proc. Priv. Enhancing Technol.*, 2019(3):6–25, 2019. URL: https://doi.org/10.2478/popets-2019-0034, doi:10.2478/POPETS-2019-0034.

[Has70] W K Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[HKE12] Yan Huang, Jonathan Katz, and David Evans. Quid-pro-quo-tocols: Strengthening semi-honest protocols with dual execution. In *2012 IEEE Symposium on Security and Privacy*, pages 272–284, 2012. doi:10.1109/SP.2012.43.

[Hop18] Sam Hopkins. *Statistical Inference and the Sum of Squares Method*. PhD thesis, Cornell University, 2018.

[IKK12] M. Saiful Islam, M. Kuzu, and M. Kantarcioglu. Access pattern disclosure on searchable encryption: Ramification, attack and mitigation. In *Network and Distributed System Security Symposium (NDSS '12)*, 2012.

[JGJS99] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[JPS21] Mireya Jurado, Catuscia Palamidessi, and Geoffrey Smith. A formal information-theoretic leakage analysis of order-revealing encryption. In *2021 IEEE 34th Computer Security Foundations Symposium (CSF)*, pages 1–16, 2021. doi:10.1109/CSF51468.2021.00046.

[JS19] Mireya Jurado and Geoffrey Smith. Quantifying information leakage of deterministic encryption. In *Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop*, CCSW'19, pages 129–139, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3338466.3358915.

[Kea93] Michael J. Kearns. The statistical query learning model. *SIAM Journal on Computing*, 22(3):679–697, 1993.

[Kel02] John Kelsey. Compression and information leakage of plaintext. In Joan Daemen and Vincent Rijmen, editors, *Fast Software Encryption*, pages 263–276, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[KF09] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge, MA, 2009.

[KKNN16] G. Kellaris, G. Kollios, K. Nissim, and A. O' Neill. Generic attacks on secure outsourced databases. In *ACM Conference on Computer and Communications Security (CCS '16)*, 2016.

[KM18] Seny Kamara and Tarik Moataz. SQL on structurally-encrypted databases. In *Advances in Cryptology–ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2–6, 2018, Proceedings, Part I 24*, pages 149–180. Springer, 2018.

[KM19] S. Kamara and T. Moataz. Computationally volume-hiding structured encryption. In *Advances in Cryptology - Eurocrypt' 19*, 2019.

[KMO18] Seny Kamara, Tarik Moataz, and Olya Ohrimenko. Structured encryption and leakae suppression. In *Advances in Cryptology - CRYPTO '18*, 2018.

[KMPP22] Evgenios M. Kornaropoulos, Nathaniel Moyer, Charalampos Papamanthou, and Alexandros Psomas. Leakage inversion. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. ACM, nov 2022. URL: https://doi.org/10.1145%2F3548606.3560593, doi:10.1145/3548606.3560593.

[KMPQ21] Seny Kamara, Tarik Moataz, Andrew Park, and Lucy Qin. A decentralized and encrypted national gun registry. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 1520–1537. IEEE, 2021.

[KMRR15] Vladimir Kolesnikov, Payman Mohassel, Ben Riva, and Mike Rosulek. Richer efficiency/security trade-offs in 2pc. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, volume 9014 of *Lecture Notes in Computer Science*, pages 229–259. Springer, 2015. doi:10.1007/978-3-662-46494-6_11.

[KMZZ20] Seny Kamara, Tarik Moataz, Stan Zdonik, and Zheguang Zhao. Opx: An optimal relational database encryption scheme. Technical report, IACR ePrint Cryptography Archive, 2020.

[KPT20] Evgenios M Kornaropoulos, Charalampos Papamanthou, and Roberto Tamassia. The state of the uniform: Attacks on encrypted databases beyond the uniform query distribution. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1223–1240. IEEE, 2020.

[KPT21] Evgenios M Kornaropoulos, Charalampos Papamanthou, and Roberto Tamassia. Response-hiding encrypted ranges: Revisiting security via parametrized leakage-abuse attacks. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 1502–1519. IEEE, 2021.

[Las01] Jean B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[LMP17] Marie-Sarah Lacharité, Brice Minaud, and Kenneth G. Paterson. Improved reconstruction attacks on encrypted data using range query leakage. *IACR Cryptology ePrint Archive*, 2017:701, 2017. URL: http://eprint.iacr.org/2017/701.

[LMP18] Marie-Sarah Lacharité, Brice Minaud, and Kenneth G. Paterson. Improved reconstruction attacks on encrypted data using range query leakage. In *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*, pages 297–314. IEEE Computer Society, 2018. doi:10.1109/SP.2018.00002.

[LS88] Steffen L Lauritzen and David Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2):157–224, 1988.

[MF06] P. Mohassel and M. Franklin. Efficiency tradeoffs for malicious two-party computation. In *Conference on Theory and Practice of Public-Key Cryptography (PKC '06)*, volume 3958 of *Lecture Notes in Computer Science*, pages 458–473. Springer, 2006.

[Mon] MongoDB [online]. URL: https://www.mongodb.com/products/queryable-encryption.

[MRR+53] N Metropolis, A W Rosenbluth, M N Rosenbluth, A H Teller, and E Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[NKW15] M. Naveed, S. Kamara, and C. V. Wright. Inference attacks on property-preserving encrypted databases. In *ACM Conference on Computer and Communications Security (CCS)*, CCS '15, pages 644–655. ACM, 2015. URL: http://doi.acm.org/10.1145/2810103.2813651, doi:10.1145/2810103.2813651.

[Par03] Pablo A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*, 95(1):1–33, 2003.

[Pea88] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco, CA, 1988.

[PKV+14] V. Pappas, F. Krell, B. Vo, V. Kolesnikov, T. Malkin, S.-G. Choi, W. George, A. Keromytis, and S. Bellovin. Blind seer: A scalable private dbms. In *Security and Privacy (SP), 2014 IEEE Symposium on*, pages 359–374. IEEE, 2014.

[PPYY19] Sarvar Patel, Giuseppe Persiano, Kevin Yeo, and Moti Yung. Mitigating leakage in secure cloud-hosted data structures: Volume-hiding for multi-maps via hashing. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 79–93. ACM, 2019. doi:10.1145/3319535.3354213.

[RC04] Christian P Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

[SDT98] M. S. Stanković, B. M. Danković, and S. B. Trićković. Some inequalities involving harmonic numbers. In G. V. Milovanović, editor, *Recent Progress in Inequalities*, volume 430 of *Mathematics and Its Applications*. Springer, Dordrecht, 1998. doi:10.1007/978-94-015-9086-0_33.

[Sha49] C. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28:657–715, 1949.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 475–484, 2014.

[Val84] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[WBMM07] Charles V. Wright, Lucas Ballard, Fabian Monrose, and Gerald M. Masson. Language identification of encrypted VoIP traffic: Alejandra y roberto or alice and bob? In *16th USENIX Security Symposium (USENIX Security 07)*, Boston, MA, August 2007. USENIX Association. URL: https://www.usenix.org/conference/16th-usenix-security-symposium/language-identification-encrypted-voip-traffic-alejandra-y.

[WP17] Charles V. Wright and David Pouliot. Early detection and analysis of leakage abuse vulnerabilities. *IACR Cryptol. ePrint Arch.*, page 1052, 2017. URL: http://eprint.iacr.org/2017/1052.

[Yao82] A. Yao. Protocols for secure computations. In *IEEE Symposium on Foundations of Computer Science (FOCS '82)*, pages 160–164. IEEE Computer Society, 1982.

[ZKMZ21] Zheguang Zhao, Seny Kamara, Tarik Moataz, and Stan Zdonik. Encrypted databases: From theory to systems. In *Conference on Innovative Data Systems Research (CIDR '21)*, 2021.