Frequency-smoothing encryption: preventing snapshot attacks on deterministically-encrypted data

Marie-Sarah Lacharité
marie-sarah.lacharite.2015@rhul.ac.uk

Kenneth G. Paterson
kenny.paterson@rhul.ac.uk

November 2, 2017

Abstract

Naveed, Kamara, and Wright (CCS 2015) applied classical frequency analysis to carry out devastating inference attacks on databases in which the columns are encrypted with deterministic and order-preserving encryption. In this paper, we propose another classical technique, homophonic encoding, as a means to combat these attacks. We introduce and develop the concept of frequency-smoothing encryption (FSE) which provably prevents inference attacks in the snapshot attack model, wherein the adversary obtains a static snapshot of the complete encrypted database, while preserving the ability to efficiently make encrypted queries to the database. We provide provably secure constructions for FSE schemes, and we empirically assess their security for concrete parameters by evaluating them against real data. We show that frequency analysis attacks (and optimal generalisations of them for the FSE setting) no longer succeed. Finally, we discuss extending our schemes to take advantage of the full generality and power of our stateful FSE framework.

1 Introduction

Deterministic Encryption (DE) is an attractive option for encrypting databases because it is equality-preserving: finding an exact match for a specific datum is just as easy as finding an exact match for its encryption. This makes it possible for a user to query a remote encrypted database using an encrypted search term, with the database server identifying matches in the encrypted domain and returning the encrypted matching records to the user. Similarly, deterministic Order-Preserving Encryption (OPE) allows users to perform efficient range searches on encrypted data. DE and OPE schemes have been widely deployed in the industry for protecting databases in this way [20, 6].

On the other hand, classical frequency analysis is a powerful attack against deterministically encrypted data: repetitions in the plaintext show up as repetitions in the ciphertext. If the plaintext distribution is not uniform and an adversary has a reference dataset from which it can compute expected plaintext frequencies, then the adversary, given access to a snapshot of the database, can match frequencies in the encrypted domain with those in the plaintext domain. In this way, it identifies which ciphertext corresponds to which plaintext. Such an attack does not target the encryption key, but instead infers plaintext information using statistical techniques. This kind of *inference attack* was recently used to great destructive effect in the work of Naveed *et al.* [20]: they were able to correctly infer large amounts of patient information from deterministically encrypted hospital records. Their work and related papers investigating leakage in DE and OPE [12, 6, 8, 23, 29, 7, 15, 10] have severely dented both the industry's and the research community's confidence in its ability to adequately protect encrypted databases whilst preserving query capabilities.

Only recently have researchers begun to investigate how to mitigate attacks based on frequency analysis. Kerschbaum [16] presented a frequency-hiding OPE scheme. The scheme does not leak any frequency information since it forbids repetition of ciphertexts. However, it has large client-side storage requirements and, because of its order-preserving nature, is vulnerable to partial plaintext recovery attacks in a snapshot attack model [10]. The SPLASHE component of the Seabed system of [22] attempts to smooth frequencies by introducing extra database columns and spurious entries. Unfortunately, SPLASHE has limited applicability and results in a 10x storage overhead in practice. We discuss this and other related work in greater detail in Section 7.

Given the importance of the problem, and the current paucity of solutions, we set out to develop rigorous means of preventing inference attacks for encrypted databases. Frequency analysis is a venerable attack method, so it is fitting that we consider a technique that is almost as old to counter it: homophonic encoding. The goal of homophonic encoding (or homophonic substitution) is to flatten the frequency distribution of ciphertexts, so that frequency analysis becomes ineffective. This is done by using encryption schemes that map each plaintext to multiple possible ciphertexts, with the number of *homophones* for each plaintext m ideally being proportional to the frequency of m. Then, in our context, an encrypted database would still contain repetitions, but every ciphertext would occur roughly equally often; frequency information would be of no use to a snapshot adversary who has a complete copy of the encrypted database. Homophonic encoding has a long history which is well documented in, for example, [13]. However, as far as we can ascertain, it appears to have received little formal analysis. Moreover, it is usually applied in contexts where adjacent data items are not independent of one other—for example, letters or words in natural language—which renders it vulnerable to attacks based on analysis of bi-grams rather than single-letter frequencies. This inherent weakness does not arise in database encryption, where each column of the database is encrypted under a separate key and entries in adjacent rows are not correlated.

Using homophonic encoding as we do leads to encryption schemes that are randomised: each encryption of a given m can (and should) result in any one of its ciphertext homophones being selected. This would seem to make performing encrypted queries on such databases impossible, since now we have many possible homophones to match on. The solution is simple: ensure that there are enough homophones to combat frequency analysis, but not so many that they cannot all be computed on the fly and sent to the database for comparison with the relevant column of ciphertexts. The question is then whether this trade-off between preventing leakage (via frequency analysis) and increasing query complexity (because of needing to match on homophones) is beneficial, providing schemes that are both secure against snapshot attackers and reasonably efficient. In the sequel, we show that the answer to this question is positive, at least for certain types of data.

However, we must immediately issue some important *caveats*. In the current work, we achieve security against only two forms of attack. The first is inference attacks made by a snapshot attacker on a per-column basis. The second is security in a "somewhat randomised" generalisation of the standard security notion for deterministic encryption due to Rogaway and Shrimpton [28]. Our security proofs and empirical evaluations are focused on these notions. We do not defend against more advanced forms of attack, such as attacks based on analysis of queries, as in [6, 8], or attacks based on correlations between columns [7]. Concretely, without some kind of query padding or query batching, it is possible to carry out frequency analysis on the *queries* made in our schemes, since the number of queries required for a given plaintext m will turn out to be roughly proportional to the frequency of m in our approach. In addition, Grubbs *et al.* recently pointed out the artificiality of the snapshot attack model [9]. Database management systems often store additional information that an attacker would capture in its snapshot, e.g. prior queries. Nevertheless, resisting snapshot attacks is necessary for achieving meaningful security in any realistic threat model, and our approach at least achieves this.

Thus, despite some limitations, we believe that our work has significant value: currently, there are few good solutions that address *any* of the recent and severe inference attacks, and we show that at least some forms of attack can be effectively combatted at low cost. We consider that our work could form the basis for a more complete solution to the problem of preventing inference attacks on encrypted databases.

1.1 Detailed technical contributions

We introduce the concept of *frequency-smoothing encryption* (FSE) which generalises (symmetric) deterministic encryption to the setting of "somewhat randomised" encryption, where each message has a relatively small number of possible ciphertexts (homophones). Our definition of frequency-smoothing encryption is general enough to capture schemes that handle message distributions that are initially unknown or that change over time. We also show how FSE supports equality queries and database joins.

We provide two security notions for FSE in Section 2. The first, called frequency-smoothing security, prevents frequency analysis attacks by requiring that a column in an encrypted database of N entries should be indistinguishable from random data (in a sense to be made precise). The second, called privacy, generalises the symmetric deterministic encryption security notion [28]. We carefully motivate our definitional choices in the main body.

We then give, in Section 3, a generic construction for FSE from any Deterministic Encryption (DE) scheme and any Homophonic Encoding (HE) scheme. The latter is a keyless primitive that transforms plaintext data using a non-deterministic encoding step, flattening the frequency distribution, resulting in a distribution that is suitable for subsequent encryption using a DE scheme. Essentially, the flattening property of the HE scheme ensures that the resulting FSE scheme is frequency smoothing, while the privacy of the DE scheme ensures the overall privacy of the FSE scheme. We also give a direct construction of an FSE scheme from an HE scheme, a PRF, and any IND\$-CPA secure encryption scheme. This construction has the advantage that decryption avoids a potentially expensive decoding step; see Section 5.

We go on to propose two simple, easy-to-implement HE schemes in Section 4. We do not claim that these schemes are novel, but nor have we found them in the literature. Both HE schemes are tunable in the sense that the amount r of randomness injected during encryption can be controlled, giving trade-offs between query efficiency and resistance to frequency analysis attacks. We are able to show, using a novel application of Kullback-Leibler (KL) divergence and based on a framework for optimal distinguishers [3], that our HE schemes asymptotically achieve perfect flattening in a statistical sense—even for computationally unbounded adversaries. However, to obtain an effective bound requires large values of r, which in turn results in high query complexity.

Given this limitation, we also carry out an empirical analysis of the effectiveness of our FSE schemes when r is moderate, evaluating them against attacks which attempt to identify plaintexts with ciphertexts via frequency analysis, in the same way as Naveed *et al.* [20]. This form of attack asks more of the adversary than is required by our frequency-smoothing security definition, so security here offers a weaker guarantee than our formal definition, but one that is pragmatically useful. This evaluation is in Section 6.

The evaluation requires us to obtain an equivalent of frequency analysis for FSE schemes, in which each plaintext can have multiple homophones in the ciphertext space. We do so using the method of Maximum Likelihood Estimation, deriving an efficient algorithm which is statistically optimal in assigning ciphertexts to possible plaintexts, in the same way that frequency analysis is—that is, by maximising the statistical likelihood of the selected assignment, cf. [17]. We believe this algorithm to itself be novel.

We then apply this algorithm on FSE-encrypted data, using the same medical dataset as was employed in [20], and the same metric of success, this being the number of hospitals in which a certain fraction of records of a given type were successfully recovered by a frequency analysis attack. In short, we show that FSE is successful in defeating our generalised version of frequency analysis for many data types, even while maintaining moderate query complexity. Indeed, the success rate of the MLE adversary is usually quickly reduced to that of a pure guessing strategy.

Section 7 contains an extended discussion of related work, while Section 8 gives our conclusions and ideas for future work.

1.2 Terminology and notation

Let D be any probability distribution on a set of messages $\mathcal{M}$. We write $f_{\mathsf{D}}(m)$ for the probability mass function of a particular message $m \in \mathcal{M}$ according to the distribution D, so $0 \leq f_{\mathsf{D}}(m) \leq 1$ for all $m \in \mathcal{M}$. The corresponding cumulative density function (cdf) is $F_{\mathsf{D}} : \mathcal{M} \to [0, 1]$, where $F_{\mathsf{D}}(m_j) = \sum_{i=1}^{j} f_{\mathsf{D}}(m_i)$ for some ordering of the messages in $\mathcal{M}$. (This ordering may be the natural one if the data is numerical; otherwise it can be arbitrary.) The support $\mathsf{support}(\mathsf{D})$ is the subset of $\mathcal{M}$ for which the pmf is non-zero. When a data owner or an adversary must guess or estimate the data's true distribution D, we use $\tilde{\mathsf{D}}$ for the owner's approximation and $\hat{\mathsf{D}}$ for the adversary's approximation.

We use $\|$ to denote concatenation. $\mathsf{Trunc}\,(x, n)$ denotes truncating the bitstring x to a length of n bits, removing the bits from the right. $\lfloor x \rceil$ denotes the integer nearest to x. When the fractional part of x is 0.5, it is always rounded up. Note that the default rounding behaviour in programming languages such as Python is bankers' rounding, where numbers whose fractional part is 0.5 are rounded to the nearest even integer.

Our analysis involves various distributions – for instance, the data's actual distribution, and what the data owner or the adversary predict the data's distribution to be. Table 1 provides an overview of our notation for these various distributions.

Table 1: Overview of our notation for various distributions.

Symbol	Domain	Description
$\tilde{\mathsf{D}}$	$\mathcal{M}$	owner's guess of the data's distribution
$\hat{\mathsf{D}}$	$\mathcal{M}$	adversary's guess of the data's distribution
D	$\mathcal{M}$	data's actual distribution
D_s	$\mathcal{E}$	encoded data's distribution for an HE or FSE scheme when state is s (introduced in Sec. 4.1)

2 Frequency-smoothing encryption (FSE)

Our definition for FSE schemes is a stateful one. Statefulness allows a powerful feature: handling initially unknown distributions. The KeyGen algorithm accepts as input an estimate $\tilde{\mathsf{D}}$ of the messages' distribution (or, say, the uniform distribution if it is unknown). If the precise distribution of the messages is known, then the state does not need to be updated when encrypting messages and the following definition simplifies accordingly.

We make the following assumptions. First, we assume that the *support* of the distribution is known even if the exact distribution is not. Second, we assume that the messages are sampled independently and are identically distributed. If the the distribution changes over time, the estimated distribution $\tilde{\mathsf{D}}$ given as input to KeyGen would need to be replaced with a set of conditional distributions describing a stochastic process. We leave this generalization as important future work.

Definition 1. *A frequency-smoothing encryption (FSE) scheme* FSE *is a triple of algorithms* FSE = (KeyGen, Encrypt, Decrypt) *such that:*

- KeyGen : $\{0,1\}^* \times \mathcal{D}_\mathcal{M} \to \mathcal{K} \times \mathcal{S}$ *takes a security parameter* $\lambda \in \{0,1\}^*$ *and a distribution* $\tilde{\mathsf{D}} \in \mathcal{D}_\mathcal{M}$ *as input and outputs a secret key* $\mathsf{sk} \in \mathcal{K}$ *and a state* $\mathsf{s} \in \mathcal{S}$ *that includes a description of the distribution* $\tilde{\mathsf{D}}$ *and maybe other information, but does not depend on the choice of* sk*.*

- Encrypt : $\mathcal{K} \times \mathcal{M} \times \mathcal{S} \to \mathcal{C} \times \mathcal{S}$ *takes a key* $\mathsf{sk} \in \mathcal{K}$*, a message* $m \in \mathcal{M}$*, and a state* $\mathsf{s} \in \mathcal{S}$ *as input and outputs a ciphertext* $c \in \mathcal{C}$ *and an updated state* $\mathsf{s}' \in \mathcal{S}$*.*

- Decrypt : $\mathcal{K} \times \mathcal{C} \times \mathcal{S} \to \mathcal{M} \cup \{\bot\}$ *takes a key* $\mathsf{sk} \in \mathcal{K}$*, a ciphertext* $c \in \mathcal{C}$*, and a state* $\mathsf{s} \in \mathcal{S}$ *as input and outputs either a message* $m \in \mathcal{M}$ *or* $\bot$*.*

KeyGen and $\mathsf{Encrypt}$ are randomized algorithms, while $\mathsf{Decrypt}$ is deterministic. Note the requirement that the state output by KeyGen be independent of the choice of key sk it outputs. Formally we require that if $(\mathsf{sk}_i, \mathsf{s}_i) \leftarrow \mathsf{KeyGen}(\lambda, \tilde{\mathsf{D}})$ for $i = 1, 2$, then s_1 and s_2 are identically distributed. For a particular key sk, call a state s' *attainable* from the state s if $\mathsf{s}' = \mathsf{s}$ or if there exists a finite sequence of messages $m_1, \ldots, m_n \in \mathcal{M}^n$ such that defining $\mathsf{s}_0 := \mathsf{s}$ and then $(c_i, s_i) \leftarrow \mathsf{Encrypt}(\mathsf{sk}, m_i, \mathsf{s}_{i-1})$ for $i = 1, \ldots, n$, then $\mathsf{s}_n = \mathsf{s}'$. A frequency-smoothing scheme is *correct* for a distribution $\tilde{\mathsf{D}}$ if for all $(\mathsf{sk}, \mathsf{s}) \leftarrow \mathsf{KeyGen}(\lambda, \tilde{\mathsf{D}})$, any message $m \in \mathcal{M}$, and any state s' attainable from s, if $(c, \mathsf{s}'') \leftarrow \mathsf{Encrypt}(\mathsf{sk}, m, \mathsf{s}')$, then $\mathsf{Decrypt}(\mathsf{sk}, c, \mathsf{s}''') = m$ for any s''' attainable from s''.

Let $\mathcal{H}^{\mathsf{FSE}}_{\mathsf{sk},\mathsf{s}}(m) := \{\mathsf{Encrypt}(\mathsf{sk}, m, \mathsf{s})\}$ be the set of all possible encryptions (homophones) of the message m with a given state s and key sk, and let $\mathcal{H}^{\mathsf{FSE}}_{\mathsf{sk},\mathsf{s}} := \bigcup_{m \in \mathcal{M}} \mathcal{H}^{\mathsf{FSE}}_{\mathsf{sk},\mathsf{s}}(m)$ be the set of all possible encryptions (homophones) of messages for a given state s and key sk. We assume that the sizes of homophone sets are independent of the choice of $\mathsf{sk} \in \mathcal{K}$, so we may write $|\mathcal{H}^{\mathsf{FSE}}_{\mathsf{s}}(m)|$ for $|\mathcal{H}^{\mathsf{FSE}}_{\mathsf{sk},\mathsf{s}}(m)|$ and $|\mathcal{H}^{\mathsf{FSE}}_{\mathsf{s}}|$ for $|\mathcal{H}^{\mathsf{FSE}}_{\mathsf{sk},\mathsf{s}}|$.

Two immediate corollaries of the correctness property are that $\mathcal{H}^{\mathsf{FSE}}_{\mathsf{s}}(m) \subseteq \mathcal{H}^{\mathsf{FSE}}_{\mathsf{s}'}(m)$ for any state s' attainable from s, and that $\mathcal{H}^{\mathsf{FSE}}_{\mathsf{s}}(m_1)$ and $\mathcal{H}^{\mathsf{FSE}}_{\mathsf{s}}(m_2)$ are disjoint unless $m_1 = m_2$, in which case $\mathcal{H}^{\mathsf{FSE}}_{\mathsf{s}}(m_1) = \mathcal{H}^{\mathsf{FSE}}_{\mathsf{s}}(m_2)$.

2.1 Using FSE

To use frequency-smoothing encryption in the intended setting—on outsourced data that is queryable—the set $\mathcal{H}^{\mathsf{FSE}}_{\mathsf{sk},\mathsf{s}}(m)$ must be easy to compute or describe for any message m given a state s and key sk. This allows a SQL query containing an expression such as `WHERE attribute = x` to be rewritten as `WHERE attribute IN (x1, x2, ...)`, where the `xi`'s compose the set of x's homophones. This rewriting effectively incurs a query blow-up, with a single query for item x being converted into a more complex query for all of x's homophones. Looking ahead, the trick will be to parameterise our FSE schemes so that this blow-up is manageable whilst still preventing frequency analysis attacks on the schemes.

FSE also supports joins, which follows directly from the ability to support equality queries. A join on unencrypted data such as `FROM t1 JOIN t2 WHERE t1.a=t2.b` would instead be written as a `UNION` of join queries having the form `FROM t1 JOIN t2 WHERE t1.a IN (x1, x2, ...) AND t2.b IN (y1, y2, ...)`. There is a join query for each possible plaintext value; the `xi`'s compose its set of homophones in column `a` of table `t1` and the `yi`'s compose its set of homophones in column `b` of table `t2`.

FSE does not natively support range queries except other than by expanding a range to a set of values and thence to a larger set of homophones. This said, the specific constructions for FSE that follow can be adapted to use OPE as a component, in which case range queries can be efficiently supported. See Section 8 for more on this.

The state s of an FSE scheme is stored locally at the client, or in a proxy which transparently performs the encryption and decryption operations. Note that s will typically include an accurate representation of the message distribution, and thus FSE schemes may not be appropriate for very large message spaces. We will evaluate the client-side storage requirements of our FSE schemes as we introduce them, but typically they are on the order of $r \cdot |\mathcal{M}|$ where r is a small parameter.

2.2 Frequency smoothing security

A frequency-smoothing scheme should do what its name implies: hide the frequency of messages from an attacker with access to a collection of ciphertexts, like a column in a database table. It should also be hard to learn anything about individual plaintexts from ciphertexts without the secret key. We formalize these notions of frequency-smoothing and privacy in two security games.

The frequency-smoothing game $\mathsf{FSE}{-}\mathsf{SMOOTH}$ (Figure 1) captures the requirement that ciphertexts do not leak any information about message frequencies, by making their distribution indistinguishable from uniform. In the $b = 0$ case of this game, the challenger uses an estimated distribution $\tilde{\mathsf{D}}$ (corresponding to a data owner's guess of its data's distribution) to initialize the state and then encrypts messages sampled according to some distribution D. In the $b = 1$ case, the challenger samples ciphertexts uniformly at random from a set having the size of what would be the homophone set if the data's true distribution D had been known from the start. The adversary receives N ciphertexts, the distribution $\tilde{\mathsf{D}}$ that the challenger uses to initialize the state when $b = 0$, and an estimate of the data's distribution $\hat{\mathsf{D}}$ (possibly different from $\tilde{\mathsf{D}}$). The adversary's goal is to distinguish these two cases. Informally, if is able to distinguish the distribution of the N ciphertexts from uniform, then the message distribution must have failed to have been smoothed by the FSE scheme.

Definition 2. *Consider the game* $\mathsf{FSE}{-}\mathsf{SMOOTH}$ *in Figure 1 in which the adversary* $\mathcal{A}$ *receives* N *ciphertexts, an estimate* $\hat{\mathsf{D}}$ *of the messages' distribution, and the distribution* $\tilde{\mathsf{D}}$ *used to initialize the state in the* $b = 0$ *case. The frequency-smoothing advantage of* $\mathcal{A}$ *against the FSE scheme* FSE *is*

$$\begin{aligned}&\mathsf{Adv}^{\mathsf{smooth}}_{\mathsf{FSE}}(\mathcal{A}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N)\\ &= 2 \cdot \left|\Pr\left[\mathsf{FSE}{-}\mathsf{SMOOTH}^{\mathcal{A},\tilde{\mathsf{D}},\hat{\mathsf{D}},\mathsf{D},N}_{\mathsf{FSE}}(\lambda) \Rightarrow 1\right] - \frac{1}{2}\right| .\end{aligned}$$

Definition 3. *An FSE scheme* FSE *is* $(\alpha, t, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N)$-$\mathsf{SMOOTH}$ *if for all adversaries* $\mathcal{A}$ *running in time at most* t *and receiving at most* N *samples,*

```
Game $\mathsf{FSE}{-}\mathsf{SMOOTH}_{\mathsf{FSE}}^{\mathcal{A},\tilde{\mathsf{D}},\hat{\mathsf{D}},\mathsf{D},N}(\lambda)$
$b \leftarrow_{\$} \{0,1\}$
if $b = 0$ then
    $(\mathsf{sk}, \mathsf{s}_0) \leftarrow \mathsf{FSE.KeyGen}(\lambda, \tilde{\mathsf{D}})$
    $m_1, \ldots, m_N \leftarrow_{\mathsf{D}} \mathcal{M}$
    for $i$ in $\{1, \ldots, N\}$ do
        $(\mathsf{s}_i, c_i) \leftarrow \mathsf{FSE.Encrypt}(\mathsf{sk}, m_i, \mathsf{s}_{i-1})$
    endfor
else
    $(\mathsf{sk}^*, \mathsf{s}_0^*) \leftarrow \mathsf{FSE.KeyGen}(\lambda, \mathsf{D})$
    $Y \leftarrow_{\$} \mathcal{C}, |Y| = |\mathcal{H}_{\mathsf{s}_0^*}^{\mathsf{FSE}}|$
    $c_1, \ldots, c_N \leftarrow_{\$} Y$
endif
$b' \leftarrow \mathcal{A}(c_1, \ldots, c_N, \tilde{\mathsf{D}}, \hat{\mathsf{D}})$
return $(b' = b)$
```

Figure 1: The frequency-smoothing game for an FSE scheme.

it holds that

$$\mathsf{Adv}_{\mathsf{FSE}}^{\mathsf{smooth}}(\mathcal{A}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N) \leq \alpha.$$

From the definition of the $\mathsf{FSE}{-}\mathsf{SMOOTH}$ game, some necessary conditions are immediately obvious: first, for an FSE scheme to be $\mathsf{FSE}{-}\mathsf{SMOOTH}$ for arbitrary $\tilde{\mathsf{D}}$ and D, the total number of homophones would always need to be large—about the number of samples, N—to handle the case where the distribution is extremely unimodal. Therefore, for more efficient constructions, it makes sense to consider schemes that are $\mathsf{FSE}{-}\mathsf{SMOOTH}$ for classes of distributions D and $\tilde{\mathsf{D}}$ that are "close enough".

Second, the size of a message m's homophone set, $|\mathcal{H}_{\mathsf{s}}^{\mathsf{FSE}}(m)|$, should be proportional to its actual frequency according to D, $f_{\mathsf{D}}(m)$. This is a consequence of the distribution over the set of all homophones being indistinguishable from uniform and each homophone corresponding to exactly one message.

The $\mathsf{FSE}{-}\mathsf{SMOOTH}$ security notion is comprehensive; it captures the possibility that the attacker has different information ($\hat{\mathsf{D}}$) about the messages' actual distribution (D) than the data owner used to initialize the state ($\tilde{\mathsf{D}}$). It also captures the possibility that the adversary has information about what the data owner estimated the data's distribution to be ($\tilde{\mathsf{D}}$). In general, the adversary may not know exactly what distribution the data owner used to initialize the state, but we assume for simplicity that it does—such an adversary is more powerful.

An important case is when the data's distribution is known by both the data owner and the attacker. In Section 4, we present schemes that are provably secure when $\mathsf{D} = \tilde{\mathsf{D}}$ (regardless of the adversary's knowledge $\hat{\mathsf{D}}$), while in Section 6, we present results of an empirical analysis of security when $\mathsf{D} = \tilde{\mathsf{D}} = \hat{\mathsf{D}}$ and compare it to security of DE when $\hat{\mathsf{D}} = \mathsf{D}$ and $\hat{\mathsf{D}} \approx \mathsf{D}$.

2.3 Message privacy

It is not enough for an FSE scheme to hide the frequencies of the messages: even if the ciphertext distribution is uniform, the adversary could still be able to decrypt messages. For example, consider the toy FSE scheme that "encrypts" messages simply by appending bitstrings to them, with the number of different appended strings being proportional to the frequency of the message; such a scheme would satisfy Definition 3, but an adversary could simply truncate the "ciphertexts" to recover plaintexts. Thus frequency smoothing alone is not sufficient for security and we also need a message privacy notion.

To obtain our message privacy definition, we adapt the deterministic privacy ("detPriv") security notion for deterministic encryption (DE) schemes [28] to our setting. That definition is itself is an adaptation of the indistinguishability-from-random-bits ("IND\$") notion of security for a nonce-based symmetric encryption scheme [26]. It is also similar to the notion of message privacy we use for deterministic encryption schemes in Section 3.2.

In the detPriv game [28], the adversary is tasked with distinguishing real encryptions of messages m of its choice from random bit-strings selected from the ciphertext space. Our $\mathsf{FSE}{-}\mathsf{PRIV}$ game diverges from the detPriv game in two ways. First, we restrict the adversary to requesting encryptions of messages sampled according to the distribution D, so the challenger can sample the messages on its behalf. Second, we allow the adversary to receive (potentially different) encryptions of the same message. In the deterministic setting, it was assumed without loss of generality that the adversary does not repeat any encryption queries since repeated encryptions would have revealed nothing new. In our setting, the encryption algorithm is probabilistic, so we allow repeated encryptions of m, but ensure they are either real encryptions or sampled from a randomly selected set Y_m of the appropriate size, that is, of size $|\mathcal{H}_{\mathsf{s}}^{\mathsf{FSE}}(m)|$.

In the $\mathsf{FSE}{-}\mathsf{PRIV}$ game in Figure 2, the challenger either initializes the state using the estimated distribution $\tilde{\mathsf{D}}$ and then encrypts messages sampled according to D, or it samples sets Y_m of the "right" size for each message m if the true distribution D had been known from the start. The adversary $\mathcal{A}$ receives N plaintext-ciphertext pairs, the estimated distribution $\hat{\mathsf{D}}$, and the distribution $\tilde{\mathsf{D}}$ the challenger uses to initialize the state when $b = 0$. It must determine how the plaintext-ciphertext pairs were generated.

```
Game $\mathsf{FSE{-}PRIV}^{\mathcal{A},\tilde{\mathsf{D}},\hat{\mathsf{D}},\mathsf{D},N}_{\mathsf{FSE}}(\lambda)$
$b \leftarrow_{\$} \{0,1\}$
$m_1, \ldots, m_N \leftarrow_{\mathsf{D}} \mathcal{M}$
if $b = 0$ then
  $(\mathsf{sk}, \mathsf{s}_0) \leftarrow \mathsf{FSE.KeyGen}(\lambda, \tilde{\mathsf{D}})$
  for $i$ in $\{1, \ldots, N\}$ do
    $(c_i, \mathsf{s}_i) \leftarrow \mathsf{FSE.Encrypt}(\mathsf{sk}, m_i, \mathsf{s}_{i-1})$
  endfor
else
  $(\mathsf{sk}^*, \mathsf{s}_0^*) \leftarrow \mathsf{FSE.KeyGen}(\lambda, \mathsf{D})$
  $Y \leftarrow_{\$} \mathcal{C}, |Y| = |\mathcal{H}^{\mathsf{FSE}}_{\mathsf{s}_0^*}|$
  for $i$ in $\{1, \ldots, N\}$ do
    if $\exists\ j < i : m_i = m_j$ do
      $Y_{m_i} := Y_{m_j}$
    else
      $Y_{m_i} \leftarrow_{\$} Y, |Y_{m_i}| = |\mathcal{H}^{\mathsf{FSE}}_{\mathsf{s}_0^*}(m_i)|$
      $Y := Y - Y_{m_i}$
    endif
    $c_i \leftarrow_{\$} Y_{m_i}$
  endfor
endif
$b' \leftarrow \mathcal{A}((m_1, c_1), \ldots, (m_N, c_N), \tilde{\mathsf{D}}, \hat{\mathsf{D}})$
return $(b' = b)$
```

Figure 2: The privacy game for an FSE scheme.

Definition 4. *Consider the message privacy game* FSE−PRIV *in Figure 2 in which the adversary receives N plaintext-ciphertext pairs, an estimate $\hat{\mathsf{D}}$ of the messages' distribution, and the distribution $\tilde{\mathsf{D}}$ used to initialize the state in the $b = 0$ case. The message-privacy advantage of $\mathcal{A}$ against the FSE scheme* FSE *is*

$$\begin{aligned}&\mathsf{Adv}^{\mathsf{priv}}_{\mathsf{FSE}}(\mathcal{A}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N)\\ &= 2 \cdot \left| \Pr\left[\mathsf{FSE{-}PRIV}^{\mathcal{A},\tilde{\mathsf{D}},\hat{\mathsf{D}},\mathsf{D},N}_{\mathsf{FSE}}(\lambda) \Rightarrow 1\right] - \frac{1}{2} \right| .\end{aligned}$$

Definition 5. *An FSE scheme* FSE *is $(\alpha, t, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N)$-*PRIV *if for all adversaries $\mathcal{A}$ running in time at most t and receiving at most N plaintext-ciphertext pairs, it holds that $\mathsf{Adv}^{\mathsf{priv}}_{\mathsf{FSE}}(\mathcal{A}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N) \leq \alpha$.*

From this definition, one necessary condition is immediately obvious: the sizes of the final homophone sets in the $b = 0$ case, $|\mathcal{H}^{\mathsf{FSE}}_{\mathsf{s}_N}(m)|$, must equal the sizes of the homophone sets in the $b = 1$ case, $|Y_m| = |\mathcal{H}^{\mathsf{FSE}}_{\mathsf{s}^*_0}(m)|$.

Recall that in the smoothness game (Figure 1), the adversary sees only ciphertexts. Frequency smoothness enforces that the sizes of each message's homophone set must be proportional to that message's frequency. In the message privacy game (Figure 2), the adversary sees plaintext-ciphertext pairs. Message privacy enforces that there is no link between plaintexts and ciphertexts except what is necessary for correctness. Both conditions are necessary for a secure frequency-smoothing scheme. In the next section, we present constructions for FSE that reflect this two-part approach.

3 Building FSE from HE and DE

One approach to building a frequency-smoothing encryption scheme is to first probabilistically encode the messages in a way that smooths the plaintext distribution, then deterministically encrypt them. In this section, we present such a two-part, modular construction that composes homophonic encoding (to smooth the frequencies) with deterministic symmetric-key encryption (to provide privacy). Sections 3.1 and 3.2 present definitions for homophonic encoding and deterministic encryption schemes respectively, while Section 3.3 describes how to compose them to get an FSE scheme.

3.1 Homophonic encoding

We consider stateful encoding schemes that are given an estimated distribution of the messages as input.

Definition 6. *A (stateful) homophonic encoding scheme* HE *is a triple of algorithms* $(\mathsf{Setup}, \mathsf{Encode}, \mathsf{Decode})$ *such that:*

- $\mathsf{Setup} : \{0,1\}^* \times \mathcal{D}_{\mathcal{M}} \to \mathcal{S}$ *is a probabilistic algorithm that takes a configuration parameter* $\lambda \in \{0,1\}^*$ *and an estimate distribution* $\tilde{\mathsf{D}}$ *over* $\mathcal{M}$ *as input and outputs some state* $\mathsf{s} \in \mathcal{S}$ *that includes a description of the distribution* $\tilde{\mathsf{D}}$ *and any other scheme parameters.*

- $\mathsf{Encode} : \mathcal{M} \times \mathcal{S} \to \mathcal{E} \times \mathcal{S}$ *is a probabilistic algorithm that takes a message* $m \in \mathcal{M}$ *and a state* $\mathsf{s} \in \mathcal{S}$ *as input and outputs an encoded message* $e \in \mathcal{E}$ *and an updated state* $\mathsf{s}' \in \mathcal{S}$.

- $\mathsf{Decode} : \mathcal{E} \times \mathcal{S} \to \mathcal{M} \cup \{\perp\}$ *is a deterministic algorithm that takes an encoded message* $e \in \mathcal{E}$ *and a state* $\mathsf{s} \in \mathcal{S}$ *as input and outputs a message* $m \in \mathcal{M}$ *or* $\perp$.

We emphasize that all algorithms and parameters in a homophonic encoding scheme are keyless, and therefore provide no message privacy.

Let $\mathcal{H}_{\mathsf{s}}^{\mathsf{HE}}(m) := \{\mathsf{Encode}(\mathsf{s}, m)\}$ be the set of all possible encodings (homophones) of the message $m \in \mathcal{M}$ for a given state s, and let $\mathcal{H}_{\mathsf{s}}^{\mathsf{HE}} := \bigcup_{m \in \mathcal{M}} \mathcal{H}_{\mathsf{s}}^{\mathsf{HE}}(m)$. In order to use HE for its intended purpose, we require that the set of homophones of a message is easy to compute or describe given a state.

Again, call a state s' *attainable* from the state s if $\mathsf{s}' = \mathsf{s}$ or there exists some finite sequence of messages $m_1, \dots, m_n \in \mathcal{M}^n$ such that setting $\mathsf{s}_0 := \mathsf{s}$ and letting $(e_i, \mathsf{s}_i) \leftarrow \mathsf{Encode}(m_i, \mathsf{s}_{i-1})$ for $i = 1, \dots, n$, then we have $\mathsf{s}_n = \mathsf{s}'$.

A homophonic encoding scheme is *correct* for a distribution $\tilde{\mathsf{D}} \in \mathcal{D}_{\mathcal{M}}$ if for all states s output by $\mathsf{Setup}(\lambda, \mathsf{D})$, any message $m \in \mathcal{M}$, and any state s' attainable from s, if $(e, \mathsf{s}'') \leftarrow \mathsf{Encode}(m, \mathsf{s}')$, then it holds that $\mathsf{Decode}(e, \mathsf{s}''') = m$ for any s''' attainable from s''. In particular, the correctness property requires that any two sets of homophones $\mathcal{H}_{\mathsf{s}}^{\mathsf{HE}}(m)$ and $\mathcal{H}_{\mathsf{s}}^{\mathsf{HE}}(m')$ are disjoint unless $m = m'$.

While encoding schemes can be fixed-length or variable-length, depending on whether the encoded messages $\mathcal{E}$ all have the same length, we consider only fixed-length schemes in this paper. The usual advantage of variable-length codes—their low average codeword length—is not as much of an advantage in this setting.[1]

In Figure 3, we introduce a game $\mathsf{HE{-}SMOOTH}$ for HE schemes that is similar to the $\mathsf{FSE{-}SMOOTH}$ game (Figure 1). Note that in the $b = 1$ case of the $\mathsf{FSE{-}SMOOTH}$ game, the adversary receives ciphertexts sampled uniformly at random from some set of the right size, while in the $b = 1$ case of the $\mathsf{HE{-}SMOOTH}$ game, the adversary receives ciphertexts sampled uniformly at random from the *actual* set of homophones. We also define the advantage of an adversary and the security of an HE scheme in a manner similar to the corresponding $\mathsf{FSE{-}SMOOTH}$ definitions of the previous section.

Definition 7. *Consider the game* $\mathsf{HE{-}SMOOTH}$ *in Figure 3. The frequency-smoothing advantage of* $\mathcal{A}$ *against the homophonic encoding scheme* HE *is*

$$\begin{aligned}&\mathsf{Adv}_{\mathsf{HE}}^{\mathsf{smooth}}(\mathcal{A}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N) \\ &= 2 \cdot \left| \Pr\left[\mathsf{HE{-}SMOOTH}_{\mathsf{HE}}^{\mathcal{A}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N}(\lambda) \Rightarrow 1 \right] - \frac{1}{2} \right| .\end{aligned}$$

[1] In a database table, it is likely that every value in a column is allocated the same amount of storage according to the declared data type of the attribute. Variable-length entries are still possible, however. For example, the MySQL version 5.7 reference manual describes four variable-length data types, all for strings: `VARCHAR`, `VARBINARY`, `BLOB` and `TEXT` [21]. Values in a `VARCHAR` column, for example, are stored with a prefix indicating their length in bytes. While the maximum length of an entry in the column must be specified, the data is not padded. Since we are considering applications where the data items are no longer than a few bytes, it is space-efficient to pad data to a fixed size and omit the length prefix.

```
Game $\mathsf{HE{-}SMOOTH}_{\mathsf{HE}}^{\mathcal{A},\tilde{\mathsf{D}},\hat{\mathsf{D}},\mathsf{D},N}(\lambda)$
$b \leftarrow_{\$} \{0,1\}$
if $b = 0$ then
    $\mathsf{s}_0 \leftarrow \mathsf{HE.Setup}(\lambda, \tilde{\mathsf{D}})$
    $m_1, \dots, m_N \leftarrow_{\mathsf{D}} \mathcal{M}$
    for $i$ in $\{1, \dots, N\}$ do
        $(e_i, \mathsf{s}_i) \leftarrow \mathsf{HE.Encode}(m_i, \mathsf{s}_{i-1})$
    endfor
else
    $\mathsf{s}_0^* \leftarrow \mathsf{HE.Setup}(\lambda, \mathsf{D})$
    $e_1, \dots, e_N \leftarrow_{\$} \mathcal{H}_{\mathsf{s}_0^*}^{\mathsf{HE}}$
endif
$b' \leftarrow \mathcal{A}(e_1, \dots, e_N, \tilde{\mathsf{D}}, \hat{\mathsf{D}})$
return $(b' = b)$
```

Figure 3: The frequency-smoothing game for an HE scheme.

Definition 8. *An HE scheme* HE *is* $(\alpha, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N)$-$\mathsf{SMOOTH}$ *if for all adversaries* $\mathcal{A}$*, it holds that*

$$\mathsf{Adv}_{\mathsf{HE}}^{\mathsf{smooth}}(\mathcal{A}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N) \leq \alpha.$$

Note that our definition of HE smoothness allows the adversary to be computationally unbounded. Our specific HE schemes in Section 4 will achieve HE smoothness in this sense.

3.2 Deterministic encryption

Deterministic encryption is the second ingredient in our modular construction for FSE schemes. We include the standard definition here for completeness.

Definition 9. *A deterministic (secret-key) encryption (DE) scheme* DE *is a triple of algorithms* $(\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *with associated sets* $\mathcal{K}$*,* $\mathcal{M}$*, and* $\mathcal{C}$ *such that:*

- $\mathsf{KeyGen} : \{0,1\}^* \to \mathcal{K}$ *is a probabilistic algorithm that takes a security parameter* λ *as input and outputs a secret key* $\mathsf{sk} \in \mathcal{K}$*.*

- $\mathsf{Encrypt} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$ *is a deterministic algorithm that takes a secret key* $\mathsf{sk} \in \mathcal{K}$ *and a message* $m \in \mathcal{M}$ *as input, and outputs a ciphertext* $c \in \mathcal{C}$*.*

- $\mathsf{Decrypt} : \mathcal{K} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\}$ *is a deterministic algorithm that takes a key* $\mathsf{sk} \in \mathcal{K}$ *and a ciphertext* $c \in \mathcal{C}$ *as input and outputs a message* $m \in \mathcal{M}$ *or* $\perp$*.*

A deterministic encryption scheme is correct if

$$\mathsf{Decrypt}(\mathsf{sk}, \mathsf{Encrypt}(\mathsf{sk}, m)) = m$$

for all $m \in \mathcal{M}$ and all $\mathsf{sk} \in \mathcal{K}$.

The security notion we choose to use for deterministic encryption is based on indistinguishability from random bits. See Figure 4. Such definitions have already been used in the context of nonce-based symmetric encryption [26] and deterministic authenticated encryption (DAE) for key-wrapping [28]. The adversary adaptively queries an encryption oracle with messages and consistently receives either the corresponding ciphertext or a string of random bits that has the same length as the ciphertext. Without loss of generality, we assume the adversary does not repeat any queries to its encryption oracle. The adversary's goal is to determine whether the oracle is responding with real ciphertexts or random bitstrings. However, to make a definition that is well-suited to the potentially small message spaces we will encounter in our FSE schemes, we deviate from previous definitions in the literature: in the "random bits" case, we sample ciphertexts uniformly at random *without* replacement from a random ciphertext set $Y \subset \mathcal{C}$ of an appropriate size. This makes our definition closer to that of PRI-security for DAE [28, Section 8], though we dispense with the decryption oracle in that notion.

```
Game $\mathsf{DE{-}PRIV}^{\mathcal{A},N}_{\mathsf{DE}}(\lambda)$
$b \leftarrow_\$ \{0,1\}$
$\mathsf{sk} \leftarrow \mathsf{DE.KeyGen}(\lambda)$
$Y \leftarrow_\$ \mathcal{C}, |Y| = |\mathcal{M}|$
$b' \leftarrow \mathcal{A}^{ENC}$
return $(b' = b)$
```

```
$ENC(m)$
if $b = 0$ then
    $c = \mathsf{DE.Encrypt}(\mathsf{sk}, m)$
else
    $c \leftarrow_\$ Y$
    $Y = Y \setminus \{c\}$
endif
return $c$
```

Figure 4: The message privacy game for a DE scheme. We assume that $\mathcal{A}$ does not repeat queries.

Definition 10. *Consider the deterministic privacy game in Figure 4. The message privacy advantage of* $\mathcal{A}$ *against the deterministic encryption scheme*

DE *is*

$$\begin{aligned}&\mathsf{Adv}^{\mathsf{priv}}_{\mathsf{DE}}(\mathcal{A}, N)\\&= 2 \cdot \left|\Pr\left[\mathsf{DE}{-}\mathsf{PRIV}^{\mathcal{A},N}_{\mathsf{DE}}(\lambda) \Rightarrow 1\right] - \frac{1}{2}\right|.\end{aligned}$$

Definition 11. *A DE scheme* DE *is said to be* (α, t, N)*-private if for all adversaries* $\mathcal{A}$ *running in time at most* t *and making at most* N *encryption queries, it holds that* $\mathsf{Adv}^{\mathsf{priv}}_{\mathsf{DE}}(\mathcal{A}, N) \leq \alpha$*.*

A block cipher that is a PRP is easily seen to meet this definition; AES would be a good candidate. For more flexibility in selecting the message space $\mathcal{M}$, one could pad short strings and use a block cipher, or use a small-domain PRP [19, 24] or a format-preserving encryption scheme [5, 4]. For larger domains, a wide-block PRP or an encryption mode such as SIV could be used [28].

3.3 FSE from HE and DE

Now that we have defined stateful HE schemes, DE schemes, and their security, we are ready to present our modular construction for an FSE scheme.

Definition 12. *Let* $\mathsf{HE} = (\mathsf{Setup}, \mathsf{Encode}, \mathsf{Decode})$ *be a stateful homophonic encoding scheme with message space* $\mathcal{M}$ *and encoded message space* $\mathcal{E}$*. Let* $\mathsf{DE} = (\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *be a deterministic encryption scheme with key space* $\mathcal{K}$*, message space* $\mathcal{E}$*, and ciphertext space* $\mathcal{C}$*. The* ***composed FSE scheme*** $(\mathsf{HE}, \mathsf{DE}) - \mathsf{FSE}$ *is defined as follows.*

- KeyGen *takes a security parameter* $\lambda \in \{0,1\}^*$ *and a distribution* $\mathsf{D} \in \mathcal{D}_{\mathcal{M}}$ *as input. It runs* $\mathsf{DE.KeyGen}(\lambda)$ *to obtain a key* $\mathsf{sk} \in \mathcal{K}$*. It also runs* $\mathsf{HE.Setup}(\lambda, \mathsf{D})$ *to obtain an initial state* s_0*. It outputs* $(\mathsf{sk}, \mathsf{s}_0)$*.*

- $\mathsf{Encrypt}$ *takes a key* $\mathsf{sk} \in \mathcal{K}$*, a message* $m \in \mathcal{M}$*, and a state* $\mathsf{s} \in \mathcal{S}$ *as input. It runs* $\mathsf{HE.Encode}(m, \mathsf{s})$ *to obtain* (e, s')*. It then runs* $\mathsf{DE.Encrypt}(\mathsf{sk}, e)$ *to obtain a ciphertext* $c \in \mathcal{C}$*. It outputs* (c, s')*.*

- $\mathsf{Decrypt}$ *takes a key* $\mathsf{sk} \in \mathcal{K}$*, a ciphertext* $c \in \mathcal{C}$*, and a state* $\mathsf{s} \in \mathcal{S}$ *as input. It runs* $\mathsf{DE.Decrypt}(\mathsf{sk}, c)$ *to obtain a message* $e \in \mathcal{E}$ *or* $\perp$*. In the former case, it then runs* $\mathsf{HE.Decode}(e, \mathsf{s})$ *to obtain a message* $m \in \mathcal{M}$ *or* $\perp$*. It outputs* m*, or* $\perp$ *if it occurred in either step.*

When the HE scheme is frequency-smoothing and the DE scheme is message-private, the composed FSE scheme is both frequency-smoothing and private, in the senses of Definitions 3 and 5. See Appendix A for the proof of the following theorem.

Theorem 13. *Suppose that* HE *is an* $(\alpha_{\mathsf{HE}}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N)$-$\mathsf{SMOOTH}$ *homophonic encoding scheme on* $(\mathcal{M}, \mathcal{E}, \mathcal{S})$ *for some* $\tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D} \in \mathcal{D}_{\mathcal{M}}$ *and that* DE *is an* $(\alpha_{\mathsf{DE}}, t + t_{\mathsf{HE.Setup}} + N \cdot (t_{\mathsf{HE.Encode}} + t_{\mathsf{HE.Decode}}), N)$-$\mathsf{PRIV}$ *deterministic encryption scheme on* $(\mathcal{K}, \mathcal{E}, \mathcal{C})$*. Then the FSE scheme* $(\mathsf{HE}, \mathsf{DE})$-$\mathsf{FSE}$ *is*

- $(\alpha_{\mathsf{HE}} + \alpha_{\mathsf{DE}}, t, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N)$-$\mathsf{SMOOTH}$*, and*
- $(\alpha_{\mathsf{HE}} + \alpha_{\mathsf{DE}}, t, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N)$-$\mathsf{PRIV}$.

4 Some static HE schemes

Henceforth, we narrow our focus to frequency-smoothing encryption for the scenario where the data's actual distribution is known to both the data owner and the adversary (so $\tilde{\mathsf{D}} = \hat{\mathsf{D}} = \mathsf{D}$) and the homophonic encoding scheme is *static*, i.e., its state depends only on $\tilde{\mathsf{D}}$. We will write $\mathsf{Adv}^{\mathsf{smooth}}_{\mathsf{HE}}(\mathcal{A}, \mathsf{D}, N)$ for the adversary's advantage when $\tilde{\mathsf{D}} = \hat{\mathsf{D}} = \mathsf{D}$. We leave the development of schemes for more complex settings to future work, but note that the second HE scheme in this section can be made dynamic to cope with a changing distribution D.

We begin with a general result about an adversary's smoothness advantage against an HE scheme. Then, we present two concrete homophonic encoding schemes. The first one is an interval-based scheme, which we analyse in detail, and the second one is a banded scheme, which we briefly consider and compare to the first scheme. We will prove the smoothness of both schemes using the general bound we now develop.

4.1 Bounding an $\mathsf{HE}-\mathsf{SMOOTH}$ adversary's advantage

When the distribution is public and the HE scheme is static, we can reinterpret the $\mathsf{HE}-\mathsf{SMOOTH}$ game from Figure 3 in terms of the resulting distribution over the encoded message space $\mathcal{E}$. Let D_{s} be this distribution—for a static HE scheme, it depends solely on the initial state s output by $\mathsf{Setup}(\lambda, \mathsf{D})$. (For an arbitrary homophonic encoding scheme, the distribution over the encoding space will involve a stochastic process.) Since a message m's homophone is chosen uniformly at random, for each of its homophones e, $f_{\mathsf{D}_{\mathsf{s}}}(e) = \frac{f_{\mathsf{D}}(m)}{|\mathcal{H}^{\mathsf{HE}}(m)|}$.

The adversary must distinguish receiving N samples drawn according to D_{s} and N samples drawn according to the uniform distribution over the set of homophones. The following bound on an $\mathsf{HE}-\mathsf{SMOOTH}$ adversary's advantage follows directly from a result of [3] showing that the error probability of an optimal distinguisher given a number of samples from two close distributions D_0 and D_1 can be bounded in terms of the *Kullback-Leibler*

(KL) divergence of D_0 with respect to D_1:

$$\mathrm{KL}\left(\mathsf{D}_0, \mathsf{D}_1\right) := \sum_{m \in \mathcal{M}} f_{\mathsf{D}_0}(m) \cdot \log \frac{f_{\mathsf{D}_0}(m)}{f_{\mathsf{D}_1}(m)}.$$

Theorem 14. *Let* HE *be a static homophonic encoding scheme with message space* $\mathcal{M}$ *and encoded message space* $\mathcal{E}$*. Let* $\mathsf{D} \in \mathcal{D}_{\mathcal{M}}$ *be a public distribution over* $\mathcal{M}$*, and let* D_{s} *be the resulting distribution over* $\mathcal{E}$ *for a state* s *output by* $\mathsf{HE.Setup}(\lambda, \mathsf{D})$*. If* $f_{\mathsf{D}_{\mathsf{s}}}(e)$ *is close to* $1/|\mathcal{H}_{\mathsf{s}}^{\mathsf{HE}}|$ *for all encodings* $e \in \mathcal{H}_{\mathsf{s}}^{\mathsf{HE}}$*, then, for any* $\mathsf{HE-SMOOTH}$ *adversary* $\mathcal{A}$*, and for sufficiently large* N*,*

$$\mathsf{Adv}_{\mathsf{HE}}^{\mathsf{smooth}}(\mathcal{A}, \mathsf{D}, N) \leq \left| \frac{1}{2} - \Phi\left(-\sqrt{\frac{N \cdot \mathrm{KL}\left(\mathsf{D}_{\mathsf{s}}, \mathsf{U}_{|\mathcal{H}_{\mathsf{s}}^{\mathsf{HE}}|}\right)}{2}} \right) \right|$$

where $\Phi(\cdot)$ *is the cdf of the standard normal distribution.*

Note that this theorem applies even to computationally unbounded adversaries. Recall that the cdf of the standard normal distribution, Φ, equals $1/2$ at 0, so the closer $N \cdot \mathrm{KL}\left(\mathsf{D}_{\mathsf{s}}, \mathsf{U}_{|\mathcal{H}_{\mathsf{s}}^{\mathsf{HE}}|}\right)$ is to 0, the smaller is any $\mathsf{HE-SMOOTH}$ adversary's advantage. Hence, in order to establish a smoothness bound on any particular static scheme HE, it is sufficient to prove bounds on $\mathrm{KL}\left(\mathsf{D}_{\mathsf{s}}, \mathsf{U}_{|\mathcal{H}_{\mathsf{s}}^{\mathsf{HE}}|}\right)$. Finally, using the fact that the pdf of a standard normal distribution peaks at 0 with value $1/\sqrt{2\pi}$, it is easy to see that a good over-bound for $\mathsf{Adv}_{\mathsf{HE}}^{\mathsf{smooth}}(\mathcal{A}, \mathsf{D}, N)$ is given by

$$\mathsf{Adv}_{\mathsf{HE}}^{\mathsf{smooth}}(\mathcal{A}, \mathsf{D}, N) \leq \frac{1}{2\sqrt{\pi}} \cdot \sqrt{N \cdot \mathrm{KL}\left(\mathsf{D}_{\mathsf{s}}, \mathsf{U}_{|\mathcal{H}_{\mathsf{s}}^{\mathsf{HE}}|}\right)}\ . \tag{1}$$

This suggests that to make the adversary's advantage very small, we need $\mathrm{KL}\left(\mathsf{D}_{\mathsf{s}}, \mathsf{U}_{|\mathcal{H}_{\mathsf{s}}^{\mathsf{HE}}|}\right) \ll 1/N$.

We now turn to the analysis of specific static encoding schemes. For convenience in what follows, we assume that $\mathcal{M} \subseteq \{0,1\}^n$.

4.2 Interval-based homophonic encoding

Informally, interval-based homophonic encoding (IBHE) partitions the set of r-bit strings according to the distribution D: message m will be allocated an interval of about $f_{\mathsf{D}}(m) \cdot 2^r$ bitstrings. Each message will be replaced by one of its corresponding r-bit strings.

One way (others are possible) of partitioning the set of r-bit strings according to D is as follows. Suppose, without loss of generality, that the messages $\mathcal{M} = \{m_1, m_2, \ldots\}$ are ordered by increasing frequency according to D. Now, consider the cumulative distribution F_{D}. To simplify notation, let $F_{\mathsf{D}}(m_0) := 0$. Then, the homophone set of any message m_i is

$$\left\{ \lfloor 2^r \cdot F_{\mathsf{D}}(m_{i-1}) \rceil, \ldots, \lfloor 2^r \cdot F_{\mathsf{D}}(m_i) \rceil - 1 \right\},$$

where integers in this set are represented with r bits. This interval has size approximately $2^r \cdot f_{\mathsf{D}}(m_i)$, as desired. The encoding algorithm for IBHE simply selects the encoding e of m_i uniformly at random from the relevant interval.

It is clear that the encoding bitlength r must be at least $\log_2 |\mathsf{support}(\mathsf{D})|$ so each message can have at least one possible encoding. In addition, we require that r is big enough so that each message is assigned a non-empty interval using this partitioning technique. The following straightforward proposition relates a message distribution, an IBHE encoding length, and a lower bound on the number of homophones each message has.

Proposition 15. *Let $\mathcal{M} = \{m_1, m_2 \ldots\}$ be a set of messages ordered by increasing frequency according to an arbitrary distribution D whose support is $\mathcal{M}$, and let $h \geq 1$ be a positive integer. Then, when encoded with r-bit IBHE, every message $m \in \mathcal{M}$ has at least h homophones if and only if $r \geq r_{min-h}$, where*

$$r_{min-h} := \left\lceil \max_{1 \leq i \leq |\mathcal{M}|} \log_2 \frac{i \cdot h - 0.5}{F_{\mathsf{D}}(m_i)} \right\rceil$$

Proof. Let ℓ_i and r_i represent the left and right endpoints (inclusive) of message m_i's homophone set:

$$\ell_i := \lfloor 2^r \cdot F_{\mathsf{D}}(m_{i-1}) \rceil \text{ and } r_i := \lfloor 2^r \cdot F_{\mathsf{D}}(m_i) \rceil - 1,$$

so the size of message m_i's homophone set is $|\mathcal{H}^{\mathsf{HE}}(m_i)| = r_i - \ell_i + 1$. By definition, $\ell_1 = 0$ and $\ell_i = r_{i-1} + 1$ for $i = 2, \ldots, |\mathcal{M}|$.

Suppose every message in $\mathcal{M}$ has at least h homophones. This happens if and only if, for each i from 1 to $|\mathcal{M}|$, we have

$$\begin{aligned}
& & r_i &\geq h + \ell_i - 1 \\
\Leftrightarrow & & \lfloor 2^r \cdot F_{\mathsf{D}}(m_i) \rceil - 1 &\geq i \cdot h - 1 \\
\Leftrightarrow & & 2^r \cdot F_{\mathsf{D}}(m_i) &\geq i \cdot h - 0.5 \\
\Leftrightarrow & & r &\geq \log_2 \frac{i \cdot h - 0.5}{F_{\mathsf{D}}(m_i)}.
\end{aligned}$$

Since this inequality must hold for all i and r is an integer, we obtain the desired expression for r_{min-h}. □

For correctness (i.e., to ensure that no message is assigned an empty homophone set), $r \geq r_{min-1}$ is necessary and sufficient.

It is possible to obtain a simpler *sufficient* (though not necessary) condition for every message to have at least h homophones by noting that messages are ordered according to D, so $F_{\mathsf{D}}(m_i) \geq i \cdot f_{\mathsf{D}}(m_1)$. We state this useful result in the following corollary.

Corollary 16. *If messages are encoded with r-bit IBHE for some $r \geq \log_2 \frac{h}{f_{\mathsf{D}}(m_1)}$, then every message $m \in \mathcal{M}$ has at least h homophones.*

Proof. For any i from 1 to $|\mathcal{M}|$, we have

$$\log_2 \frac{h}{f_{\mathsf{D}}(m_1)} \geq \log_2 \frac{i \cdot h - 0.5}{i \cdot f_{\mathsf{D}}(m_1)} \geq \log_2 \frac{i \cdot h - 0.5}{F_{\mathsf{D}}(m_i)}.$$

□

Therefore, the condition $r \geq \log_2 \frac{h}{f_{\mathsf{D}}(m_1)}$ is enough to guarantee that all messages have at least h homophones.

Definition 17. *The interval-based homophonic encoding (IBHE) scheme with message space $\mathcal{M} \subseteq \{0,1\}^n$ is defined as follows:*

- $\mathsf{Setup} : (\lambda, \mathsf{D}) \mapsto \mathsf{s}$, *computes the maximum r of the minimum encoding length r_{min-1} and the encoding length $r_{\mathsf{D},\lambda}$ determined by D and λ, and outputs the state $\mathsf{s} := (r, \mathsf{D})$.*
- $\mathsf{Encode} : (m, \mathsf{s}) \mapsto e$, *chooses an integer e uniformly at random from the set of m's homophones $\mathcal{H}^{\mathsf{HE}}_{\mathsf{s}}(m) := \left\{ \lfloor 2^r \cdot F_{\mathsf{D}}(m_{i-1}) \rceil, \ldots, \lfloor 2^r \cdot F_{\mathsf{D}}(m_i) \rceil - 1 \right\}$, and outputs the r-bit representation of e.*
- $\mathsf{Decode} : (e, \mathsf{s}) \mapsto m$, *determines the message m_i such that $e \in \{F_{\mathsf{D}}(m_{i-1}), \ldots, F_{\mathsf{D}}(m_i) - 1\}$, and outputs $m := m_i$.*

Note that it is possible for the encoded bitlength r to be smaller than the data's bitlength n, in which case IBHE compresses data. Also note that IBHE's Encode and Decode algorithms need access to tables mapping the messages m_i to their intervals

$$\left\{ \lfloor 2^r \cdot F_{\mathsf{D}}(m_{i-1}) \rceil, \ldots, \lfloor 2^r \cdot F_{\mathsf{D}}(m_i) \rceil - 1 \right\}$$

via the cdf F_{D} of D, and *vice versa.* Since each interval can be represented by $2r$ bits, we see that the total client-side storage for these tables is $4r \cdot |\mathcal{M}|$ bits.

In order to apply Theorem 14 to bound the HE-smoothness of IBHE, and thereby Theorem 13 to construct an FSE scheme, we need an upper bound on the Kullback-Leibler divergence of the encoded data's distribution D_{s} relative to the uniform distribution $\mathsf{U}_{|\mathcal{H}^{\mathsf{HE}}_{\mathsf{s}}|}$. For IBHE, if the encoding length r is at least r_{min-h}, as defined in the statement of Prop. 15, then this bound is approximately $1/2h^2$. This result is stated in the following lemma, whose proof is in Appendix B.

Definition 18. *Let* D *be a distribution over* $\mathcal{M}$ *and suppose that* m_1 *is the least frequent message according to* D*. Suppose that the encoding length* r *in the IBHE scheme is such that* $r \geq r_{min-h}$ *for some positive integer* h *and let* $\mathsf{s} := (r, \mathsf{D})$*. Then,*

$$\mathrm{KL}\left(\mathsf{D}_{\mathsf{s}}, \mathsf{U}_{2^r}\right) \leq \frac{1}{2h^2}.$$

Suppose one has a distribution D, N samples, and a given target ϵ for the frequency-smoothing advantage $\mathsf{Adv}^{\mathsf{smooth}}_{\mathsf{HE}}(\mathcal{A}, \mathsf{D}, N)$ for the IBHE scheme. Using the approximation in eqn. 1 from the start of this section and the bound from the above lemma, we obtain after some manipulation the requirement

$$h \geq \frac{\sqrt{N}}{2\sqrt{2\pi}\epsilon}.$$

Combining this value with the sufficient condition from Cor. 16 enables us to derive a value for r to use in the IBHE scheme:

$$r \geq \log_2 \frac{\sqrt{N}}{2\sqrt{2\pi}\epsilon \cdot f_{\mathsf{D}}(m_1)}.$$

Note that to halve the upper bound on an adversary's advantage, the minimum encoding length increases by 1 bit.

A numerical example. Suppose D is such that $f_{\mathsf{D}}(m_1) = 2^{-5}$. Suppose $N = 2^{10}$ and $\epsilon = 2^{-10}$. Then we get $h \geq 2^{15}/2\sqrt{2\pi} \approx 2^{12.7}$. Applying the bound from Cor. 16 to guarantee $r \geq r_{min-h}$, we find that we need $r \geq 18$ to limit the frequency-smoothing advantage of any adversary to at most ϵ against IBHE for these parameters.

4.2.1 IBHE variants

We now describe, with practicality in mind, two variants of IBHE.

(Variant 1) Encodings are appended to messages rather than replacing them. This enables, for instance, faster decoding when processing query results.

(Variant 2) Modify how intervals (homophone sets) are allocated in such a way that smaller encoding bitlengths are possible (as long as they are still at least $\log_2 |\mathsf{support}(\mathsf{D})|$). Some distributions can yield prohibitively large values of r_{min-1} if $f_{\mathsf{D}}(m_1)$ is relatively tiny.

The change to how intervals of $\{0, \ldots, 2^r - 1\}$ are assigned can be interpreted simply as building intervals (in the same way as before) for a modified distribution D'. The algorithm shown in Figure 5 takes as input a distribution D and a desired encoding length. It outputs a second distribution, D', with the same support as D that can be used to construct intervals, encode, and decode with the desired encoding length. Starting with the least frequent message, this algorithm changes the distribution just enough that one homophone is assigned to each "too small" message. It does this until until each of the remaining messages can be assigned at least one homophone after being scaled to share the error introduced by assigning "too many" homophones to the least frequent messages. When $r \geq r_{min-1}$, this algorithm does not change the distribution.

The resulting modified IBHE scheme would run this algorithm as part of Setup and use the adjusted distribution D' in the state, $\mathsf{s} := (r, \mathsf{D}')$, for all encodings and decodings. The original distribution D does not need to be stored.

```
Distribution adjustment algorithm
$isBigEnough = False,\ maxAdj = 0,\ scaleFactor = 1$
for $i$ in$\{1, \ldots, |\mathcal{M}|\}$ do
  if $i = 1$ then
    if $f_{\mathsf{D}}(m_i) < 1/2^{r+1}$ then
      $f_{\mathsf{D}'}(m_i) = 1/2^{r+1}$
      $maxAdj = 1$
      $scaleFactor = (1 - f_{\mathsf{D}}(m_i))/(1 - f_{\mathsf{D}'}(m_i))$
    else
      $f_{\mathsf{D}'}(m_i) = f_{\mathsf{D}}(m_i)$
      // second value could still be too small
  else // $i \geq 2$
    if $isBigEnough$ then
      $f_{\mathsf{D}'}(m_i) = f_{\mathsf{D}}(m_i)/scaleFactor$
    else
      if $f_{\mathsf{D}}(m_i) \geq 1/2^r \cdot scaleFactor$ then
        $isBigEnough = True$
        $f_{\mathsf{D}'}(m_i) = f_{\mathsf{D}}(m_i)/scaleFactor$
      else
        $f_{\mathsf{D}'}(m_i) = 1/2^r$
        $maxAdj = i$
        $scaleFactor = (1 - F_{\mathsf{D}}(m_{maxAdj}))/$
              $(1 - F_{\mathsf{D}'}(m_{maxAdj}))$
return $\mathsf{D}'$
```

Figure 5: Distribution adjustment algorithm for distribution D and desired encoding length r, with $r \geq \log_2 |\mathsf{support}(\mathsf{D})|$.

4.3 Banded homophonic encoding

We next present a simple homophonic encoding scheme that appends tags to messages rather than replacing them with encodings, which the previous scheme did. The tags can have any length $l \geq 1$ and each message has at most 2^l homophones. Suppose again that the messages are ordered by increasing frequency according to the distribution D:

$$f_{\mathsf{D}}(m_1) \leq f_{\mathsf{D}}(m_2) \leq \ldots \leq f_{\mathsf{D}}(m_{|\mathcal{M}|}).$$

Based on these frequencies, each message has a *band* that determines the number of possible tags that can be appended to it and therefore the number of homophones it has. Divide the interval $(0, f_{\mathsf{D}}(m_{|\mathcal{M}|})]$ into 2^l bands each of width $\mathsf{w} := f_{\mathsf{D}}(m_{|\mathcal{M}|})/2^l$, numbered 1 to 2^l. The messages whose frequencies are in band i, in the interval $((i-1) \cdot \mathsf{w}, i \cdot \mathsf{w}]$, will all have i homophones. In particular, the most frequent message, $m_{|\mathcal{M}|}$, will have 2^l homophones—all possible l-bit strings can be appended to it.

Definition 19. *The banded homophonic encoding (BHE) scheme with message space $\mathcal{M} \subseteq \{0,1\}^n$ is defined as follows:*

- $\mathsf{Setup} : (\lambda, \mathsf{D}) \mapsto \mathsf{s}$ *computes the tag length l determined by λ and D, the band width $\mathsf{w} := f_{\mathsf{D}}(m_{|\mathcal{M}|})/2^l$, and outputs $\mathsf{s} := (l, \mathsf{w}, \mathsf{D})$.*

- $\mathsf{Encode} : (m, \mathsf{s}) \mapsto m \| t$ *computes message m's frequency band, $\mathsf{b} := \lceil f_{\mathsf{D}}(m)/\mathsf{w} \rceil$, picks an integer t uniformly at random in $\{0, 1, ..., \mathsf{b}-1\}$, and outputs the $(n+l)$-bit string $m \| t$, where t is represented using l bits.*

- $\mathsf{Decode} : (e, \mathsf{s}) \mapsto \mathsf{Trunc}\,(e, n)$ *removes the last l bits of e to recover $m := \mathsf{Trunc}\,(e, n)$.*

The main advantages of this banded HE scheme are that there is no minimum tag length and decoding is fast—in particular, it does not need any table of frequency information for D. Encoding requires storing a table of $l \cdot |\mathcal{M}|$ bits.

Another feature is that if the distribution changes, the scheme can adapt to the new frequencies without re-encoding every data item. This can be done by using so-far-unused l-bit tags if an item's frequency increases (effectively increasing its band number), or by over-sizing l to begin with and using a deliberately under-sized sets of homophones initially and, if an item's frequency decreases, re-scaling the bands used for all the other items. By contrast, the interval-based encoding scheme cannot adapt to changes in the distribution without re-encoding all of the messages.

A negative aspect of the banded homophonic encoding scheme is that the total number of encodings, $|\mathcal{H}_{\mathsf{s}}^{\mathsf{HE}}|$, is not fixed. For Theorem 14 to apply, the distribution of the encoded data must already be close enough to the uniform distribution on its homophones. Consider the rounding errors for each message: let

$$\delta_i := \left\lceil 2^l \cdot f_{\mathsf{D}}(m)/f_{\mathsf{D}}(m_{|\mathcal{M}|}) \right\rceil - 2^l \cdot f_{\mathsf{D}}(m)/f_{\mathsf{D}}(m_{|\mathcal{M}|}),$$

so $\delta_i \in [0,1)$ for each m_i, $1 \leq i \leq |\mathcal{M}|$. The total number of homophones is then

$$|\mathcal{H}_{\mathsf{s}}^{\mathsf{HE}}| = \frac{2^l}{f_{\mathsf{D}}(m_{|\mathcal{M}|})} + \sum_{i=1}^{|\mathcal{M}|} \delta_i.$$

Whereas the total number of homophones was predictable (fixed, actually) for IBHE, here it may vary by as much as $|\mathcal{M}| - 1$ depending on the distribution and the rounding errors δ_i it produces. For the encoded data's distribution to be close enough to uniform so we can apply Theorem 14, we require $|\mathcal{M}| \ll \frac{2^l}{f_{\mathsf{D}}(m_{|\mathcal{M}|})}$. This unpredictability indicates that values of l for BHE will need to be much higher than values of r for IBHE to guarantee smoothness. This is quantified in the following lemma.

Definition 20. *Let* D *be a distribution over* $\mathcal{M}$ *and suppose that* $m_{|\mathcal{M}|}$ *is the most frequent message according to* D. *Suppose that* l *in the BHE scheme is such that* $|\mathcal{M}| \ll \frac{2^l}{f_{\mathsf{D}}(m_{|\mathcal{M}|})}$, *and let* $|\mathcal{H}_{\mathsf{s}}^{\mathsf{HE}}|$ *be the size of the resulting set of homophones. Then*

$$\mathrm{KL}\left(\mathsf{D}_{\mathsf{s}}, \mathsf{U}_{|\mathcal{H}_{\mathsf{s}}^{\mathsf{HE}}|}\right) \leq \frac{|\mathcal{M}| \cdot f_{\mathsf{D}}(m_{|\mathcal{M}|})}{2^{l+1}}.$$

The proof of this Lemma is in Appendix C.

Suppose one has a distribution D, N samples, and a given target ϵ for the frequency-smoothing advantage $\mathsf{Adv}_{\mathsf{HE}}^{\mathsf{smooth}}(\mathcal{A}, \mathsf{D}, N)$ for the BHE scheme. Using the above lemma and the bound on an adversary's advantage in eqn. 1 from the start of this section, we obtain the requirement

$$l \geq \log_2 \left(\frac{N \cdot |\mathcal{M}| \cdot f_{\mathsf{D}}(m_{|\mathcal{M}|})}{(2\epsilon)^2 \cdot \pi} \right) - 1.$$

Note that since $f_{\mathsf{D}}(m_{|\mathcal{M}|})$ is the maximum frequency, $f_{\mathsf{D}}(m_{|\mathcal{M}|}) \geq \frac{1}{|\mathcal{M}|}$, so regardless of the distribution, the added bitlength l must be at least

$$\log_2 \left(\frac{N}{(2\epsilon)^2 \cdot \pi} \right) - 1.$$

A numerical example. Suppose $N = 2^{10}$ and $\epsilon = 2^{-10}$, and let D be the given distribution on the message space $\mathcal{M}$. A lower bound on the required tag length l in the BHE scheme is $\log_2 \left(\frac{2^{10}}{(2 \cdot 2^{-10})^2 \cdot \pi} \right) - 1 \approx 25$. The minimum value of l needed for a specific distribution may be greater still.

Recall the similar example at the end of Section 4.2: for the same values of N and ϵ, the minimum required encoding bitlength for interval-based HE was $r \geq 12.7 + \log_2 \frac{1}{f_{\mathsf{D}}(m_1)}$. With banded HE, the minimum *additional* bitlength is $l = 25$.

5 Building FSE from HE and CIV

While the modularity of the composed approach to achieving FSE may offer control over the security-efficiency trade-offs and choice of DE scheme, an all-in-one approach with no separate decryption and decoding steps can be more efficient. In this section, we describe how to build an FSE scheme of this type, from any HE scheme, a PRF, and an IND\$-CPA encryption scheme. This approach is somewhat inspired by the synthetic IV (SIV) construction of Rogaway and Shrimpton [27].

Definition 21. *Let* $\mathsf{HE} = (\mathsf{HE.Setup}, \mathsf{HE.Encode}, \mathsf{HE.Decode})$ *be a homophonic encoding scheme with associated spaces* $\mathcal{M}$, $\mathcal{E}$, $\mathcal{D}_{\mathcal{M}}$, *and* $\mathcal{P}$. *Let* $\mathsf{CIV} = (\mathsf{CIV.KeyGen}, \mathsf{CIV.Encrypt}, \mathsf{CIV.Decrypt})$ *be a conventional IV-based encryption scheme, as defined in [27], with key space* $\mathcal{K}_1$, *message space* $\mathcal{E}$, *IV space* $\mathcal{IV}$, *and ciphertext space* $\mathcal{C}$. *Let* PRF *be a pseudorandom function with keyspace* $\mathcal{K}_2$ *and output space* $\{0,1\}^n \subseteq \mathcal{IV}$. *The* ***SIV-like*** $(\mathsf{HE}, \mathsf{CIV}) - \mathsf{FSE}$ *scheme is defined as follows.*

- KeyGen *takes a security parameter* $\lambda \in \{0,1\}^*$ *and a distribution* $\mathsf{D} \in \mathcal{D}_{\mathcal{M}}$ *as input. It runs* $\mathsf{CIV.KeyGen}(\lambda)$ *to obtain keys* $\mathsf{sk} \in \mathcal{K}_1$ *and selects* $\mathsf{sk}_2 \leftarrow_{\$} \mathcal{K}_2$. *It also runs* $\mathsf{HE.Setup}(\lambda, \mathsf{D})$ *to obtain an encoding parameter* p. *It outputs* $(\mathsf{sk}_1, \mathsf{sk}_2, \mathsf{p})$.

- $\mathsf{Encrypt}$ *takes keys* $(\mathsf{sk}_1, \mathsf{sk}_2) \in \mathcal{K}_1 \times \mathcal{K}_2$, *an encoding parameter* $\mathsf{p} \in \mathcal{P}$, *and a message* $m \in \mathcal{M}$ *as input. First, it runs* $\mathsf{HE.Encode}(\mathsf{p}, m)$ *to obtain the encoded message* $e \in \mathcal{E}$. *It then computes* $\mathsf{PRF}(\mathsf{sk}_2, e)$ *to get* $\mathsf{iv} \in \mathcal{IV}$. *Lastly, it runs* $\mathsf{CIV.Encrypt}(\mathsf{sk}_1, m; \mathsf{iv})$ *to obtain a ciphertext* $c \in \mathcal{C}$. *It outputs* $\hat{c} = \mathsf{iv} \| c$.

- $\mathsf{Decrypt}$ *takes keys* $(\mathsf{sk}_1, \mathsf{sk}_2) \in \mathcal{K}_1 \times \mathcal{K}_2$ *and a ciphertext* $\hat{c}$ *as input. It parses* $\hat{c}$ *as* $\mathsf{iv} \| c \in \mathcal{IV} \times \mathcal{C}$. *It runs* $\mathsf{CIV.Decrypt}(\mathsf{sk}_1, c; \mathsf{iv})$ *to obtain a message* m, *and returns* m.

Notice that this scheme does not run HE.Decode during decryption, so avoiding the need to store a decoding table for HE and making it potentially more attractive for implementation.

We omit a detailed security analysis of this scheme. Its FSE-privacy follows easily from the IND\$-CPA security of CIV, noting that the use of a PRF PRF to generate the IVs from encodings e produces IVs that are indistinguishable from random, up to repetitions induced by the encoding scheme, such encodings arising only from message repetitions, and therefore resulting in identical ciphertexts $\hat{c} = \mathsf{iv} \| c$. FSE-smoothness, on the other hand, follows from the smoothness of HE, the pseudorandomness of PRF and the IND\$-CPA security of CIV.

The construction here generalises to build an FSE scheme from any HE scheme, a PRF, and any DAE scheme, in the sense introduced in [28], including SIV (though the integrity properties enjoyed by DAE are overkill for FSE in our snapshot attacker model). The idea is to set the *header* for the DAE scheme to be $\mathsf{PRF}(\mathsf{sk}_2, e)$ where $e = \mathsf{HE.Encode}(\mathsf{p}, m)$, as in the above construction.

6 Empirical assessment of FSE

In this section, we report on an empirical assessment of the security of FSE against frequency analysis attacks. Of course, we are also interested in achieving $\mathsf{FSE{-}PRIV}$, but this is easily done using our HE-DE construction with an appropriate DE component, e.g., a block cipher such as AES.

Recall that $\mathsf{FSE{-}SMOOTH}$ security is an indistinguishability-style notion designed to prevent frequency analysis attacks. However, as we have seen in numerical examples for our IBHE encoding scheme, achieving typical cryptographic security levels for this notion would require large values of r, leading to a serious blow-up in query complexity. In this section, therefore, we adopt a more pragmatic approach, working with moderate values of r and choosing as a security metric the number of data items that an attacker can correctly decrypt, as per Naveed, Kamara, and Wright [20]. Our aim is to reduce the attacker's success rate to that of a naive guessing attack. We develop a maximum likelihood attack for this setting, and then assess its performance using the same health data as was attacked in [20]. This allows us to compare the security of FSE and of DE, and of FSE to naive guessing attacks.

This approach is in line with the paradigm of *accelerated provable security* [11]: we designed a scheme and proved its security based on the security of its primitives, but we relax the primitives for practical use and rely on cryptanalysis to assess security.

6.1 Details of the approach

We work with an FSE scheme built from static HE and DE using our modular construction. For the HE component, we use IBHE (Section 4.2) with the distribution adjustment algorithm (variant 2). Our attacks on FSE are in the public distribution setting, where $\tilde{\mathsf{D}} = \hat{\mathsf{D}} = \mathsf{D}$. This grants the adversary greater power than in the scenario considered in [20], where $\hat{\mathsf{D}}$ is only approximately D.

To obtain D, we work with patient discharge data from the 200 largest hospitals in the 2009 Nationwide Inpatient Sample (NIS), from the Healthcare Cost and Utilization Project (HCUP), run by the Agency for Healthcare Research and Quality in the United States [1]. The largest hospitals were those with the greatest total number of discharges in that year. The 12 target attributes are listed in Table 2 in Appendix D. They include age in years, length of stay in days, sex, major diagnostic category, and admission type.

We simulate FSE-encrypting and then attacking the HCUP data of the *individual* largest hospitals using *each* of the hospitals' data to define a *per-hospital* reference distribution for each of the 12 target attributes. We assume this per-hospital distribution is always known to the attacker. This experimental setup is good for the attacker—in reality, it is likely that an attacker attempting to steal a particular hospital's data would only have access to, say, national statistics from previous years. That was the situation considered in [20]. To simplify our analysis, we ignore all values that were identified as missing, invalid, unavailable, or inconsistent.

6.2 A maximum likelihood attack on static FSE

Given the selected metric of success—the number of records an attacker can correctly decrypt—we must determine how an attacker would maximize this number. We apply the technique of maximum likelihood estimation (MLE) to derive an efficient attack on a static FSE scheme under the assumption that only frequency information is meaningful, thus assessing security in the $\mathsf{FSE}\text{−}\mathsf{SMOOTH}$ sense. MLE is an asymptotically optimal technique; as the number of samples tends toward infinity, the maximum likelihood estimator is an unbiased estimator with the smallest variance.

Suppose the adversary has N FSE-encrypted items, each of whose underlying plaintext was sampled independently from $\mathcal{M}$ according to the known distribution D. The adversary can compute the number of homophones $|\mathcal{H}_{\mathsf{s}}^{\mathsf{FSE}}(m)|$ for each m in $\mathcal{M}$, since this set's size depends on the state s, which in turn depends only on the distribution and not the particular choice of key.

Suppose there are $|\mathcal{M}| = |\mathsf{support}(\mathsf{D})|$ distinct plaintext items and $|\mathcal{H}^{\mathsf{FSE}}| = |\mathcal{C}|$ distinct ciphertexts, so that every possible ciphertext appears at least once. The adversary's goal is to find the correct many-to-one decryption mapping $\theta : \mathcal{C} \to \mathcal{M}$. Let $n(c)$ denote the number of times that ciphertext $c \in \mathcal{C}$ occurs in the set of samples. The attack is as follows, with Appendix E discussing the required assumptions and justification. Label the distinct observed ciphertexts so their counts are in decreasing order:

$$n(c_1) \geq n(c_2) \geq \cdots \geq n(c_{|\mathcal{C}|}).$$

Also label the $|\mathcal{M}|$ plaintext items so their scaled frequencies are in decreasing order:

$$\frac{f_{\mathsf{D}}(m_1)}{|\mathcal{H}_{\mathsf{s}}^{\mathsf{FSE}}(m_1)|} \geq \frac{f_{\mathsf{D}}(m_2)}{|\mathcal{H}_{\mathsf{s}}^{\mathsf{FSE}}(m_2)|} \geq \cdots \geq \frac{f_{\mathsf{D}}(m_{|\mathcal{M}|})}{|\mathcal{H}_{\mathsf{s}}^{\mathsf{FSE}}(m_{|\mathcal{M}|})|}.$$

Then the attack sets θ so that

$$\begin{aligned}
\theta :&\{c_1, \ldots, c_{|\mathcal{H}_{\mathsf{s}}^{\mathsf{FSE}}(m_1)|}\} \mapsto m_1, \\
\theta :&\{c_{|\mathcal{H}_{\mathsf{s}}^{\mathsf{FSE}}(m_1)|+1}, \ldots, c_{|\mathcal{H}_{\mathsf{s}}^{\mathsf{FSE}}(m_1)|+|\mathcal{H}_{\mathsf{s}}^{\mathsf{FSE}}(m_2)|}\} \mapsto m_2,
\end{aligned}$$

and so on, until all observed ciphertexts have been assigned a message.

This efficient procedure creates a decryption mapping θ that is not necessarily unique: if two or more encrypted data item counts are the same, then permuting them will result in decryption mappings that are equally likely. Similarly, if two or more scaled plaintext frequencies are the same, then permuting them will result in equally likely decryption mappings. In our experiments, such ties were broken randomly.

Notice that if deterministic encryption were used in place of FSE, so that $|\mathcal{H}_{\mathsf{s}}^{\mathsf{FSE}}(m)| = 1$ for each $m \in \mathcal{M}$, then this attack reduces to a basic frequency analysis attack of the type used in [20], which was shown to be maximum likelihood in [17]. Thus our attack generalises basic frequency analysis.

This attack is easily modified for the case where the attacker and data owners have different information about the data's distribution ($\hat{\mathsf{D}} \neq \tilde{\mathsf{D}}$). In this case, the attacker would number the plaintext items according to $f_{\hat{\mathsf{D}}}(m)/|\mathcal{H}_{\tilde{\mathsf{s}}}^{\mathsf{FSE}}(m)|$, where $\tilde{\mathsf{s}}$ depends only on $\tilde{\mathsf{D}}$.

6.3 Results

We use the aforementioned MLE attack to simulate an attacker attempting to decrypt FSE-encrypted records in a database. Our results are presented in a series of graphs in Appendix F, one for each of the attributes in Table 2, and with various encoding lengths r for each attribute. These graphs show complementary cumulative distributions, since we are interested in the number of hospitals for which *at least* some fraction of the records were recovered. We consider each attribute separately, so "percentages of records recovered" refers not to entire records (rows) in a database, but to the values of a particular attribute (column) in those records.

Our goal, informally, is that attacking FSE is hard—in particular, at least as hard as attacking DE. If our attacks are less successful against FSE than DE, then the lines corresponding to FSE will be to the left of and below those for DE, and the area under them will be smaller.

The trivial guessing attack. Of course an attacker can always simply guess that every ciphertext it sees corresponds to the most likely plaintext. It would succeed quite well with this metric for certain attributes, irrespective of the encryption method used. This is the case, for example, with `DIED` where there is one very likely plaintext (and one quite unlikely one). Each attribute's graph in Appendix F includes a solid gray line, labelled "max f_{D}", that represents the success rate of this trivial attack. No encryption method can force the trivial attacker below this line, so little security is achievable for certain attributes like `DIED` using any form of encryption (according to the metric chosen for our evaluation).

Recall that our MLE approach does not capture this trivial attack since it looks for a *correct* decryption mapping that respects the numbers of homophones each plaintext has. Thus, it is possible for the trivial attack to actually perform better than an "optimal" attack. As can be seen from the graphs, by setting r appropriately, we can ensure that this is the case, making the MLE attack worse than simple guessing. Since it is not possible for any encryption scheme to protect against simple guessing attacks, the fact that the MLE attack is made worse than the trivial attack by homophonic encoding is a positive feature of our approach. Indeed, once this is achieved for a particular value of r, there is no benefit in increasing r further (except perhaps to disguise which database column is which).

Comparison with DE. Naveed *et al.* attacked DE-encrypted 2009 data using aggregated 2004 data across the 200 largest hospitals for the auxiliary distribution [20]. The power of frequency analysis attacks on DE can be further strengthened by assuming the attacker knows the *exact* distribution on a per-hospital basis. In evaluating DE, we consider both situations, yielding two curves for DE in each graph: one from using an aggregated distribution ($\hat{\mathsf{D}} \approx \mathsf{D}$, similar to [20], but from the same year) and the other, a per-hospital distribution ($\hat{\mathsf{D}} = \mathsf{D}$). Our experiments attacking FSE always assume that the attacker has exact knowledge of the data's distribution D, giving the attacker the most power.

For some attributes, frequency analysis on DE even with aggregated data recovers nearly *all* records for *all* hospitals (e.g., `APRDRG_Risk_Mortality`, `DIED`, `FEMALE`). Frequency analysis of DE with per-hospital distributions performs even better, recovering nearly 100% of records correctly in every case. And, as can be seen from our graphs in Appendix F, FSE withstands attacks much better than DE in the majority of cases, even when the adversary is given the per-hospital distributions. The results for `AGE`, `LOS`, and `MDC` are particularly encouraging. One exception is `DIED`; using FSE barely reduces the number of records an attacker can recover, even with large encoding lengths. The reason is that `DIED` is binary and one value accounts for over 98% of records in a data set, on average. Thus the MLE attack will still succeed with high probability, as it will assign the majority of ciphertexts to the high probability value and be correct most of the time. As noted above, in such a situation, the trivial plaintext recovery attack that just assigns every ciphertext to the most likely plaintext value performs even better and is also unavoidable for *any* encryption scheme.

Limit case. As the encoding length r increases, there are fewer repeated ciphertexts, and eventually, no ciphertext will occur more than once. Recall that given N ciphertext items, the MLE attack assigns approximately $N \cdot f_{\mathsf{D}}(m)$ of them to message m. For large enough N, we can approximate this assignment of plaintexts to ciphertexts in the following manner: for each ciphertext, the attacker independently samples from $\mathcal{M}$ according to D to determine its guess. The probability that any single ciphertext is assigned the correct plaintext is then $f := \sum_{m\in\mathcal{M}} f_{\mathsf{D}}(m)^2$, and the number of correct guesses then follows a binomial distribution with N trials and success probability f. We have simulated such an attack strategy using each individual hospital's distribution and indicated the resulting curves with $r \to \infty$ in the graphs. The fraction of records recovered quickly converges to this random guessing strategy, even using encoding bitlengths much less than the values of r_{min}.

Success when using distribution adjustment algorithm. Recall variant 2 of our IBHE scheme: when the desired encoding length is less than r_{min}, intervals are constructed in a different way that guarantees even the least frequent items have at least one homophone. The values of r_{min} were highest for `AGE` (20) and `LOS` (23). Using an encoding length of 8 for `AGE` still resulted in fewer records decrypted than with DE. For `LOS`, whose minimum *unencoded* bitlength is 9, there was a drastic drop in the percentage of records recovered even with an encoding length of only 10. Using only DE, 50% of hospitals had at least 80% of their records recovered, while with 10-bit IBH encoding, no hospital had more than 22% of its records recovered.

Query complexity. For large enough encoding lengths r, our results indicate that this statistically optimal MLE attack offers no advantage over guessing—even when the attacker has precise knowledge of the underlying data's distribution. However, the parameter r affects query complexity in addition to storage cost: an equality query for one item becomes an equality query for each of its homophones. Nevertheless, the results quickly converge to random guessing for all attributes, and the effect on query complexity is manageable. For example, encoding `AGE` with $r = 10$ bits results in a query expansion of $2^r \cdot f_{\mathsf{D}}(0) \approx 2^7$ in the worst case (for the most frequent age, 0). Encoding `MDC` with $r = 10$ bits results in a query expansion of about 2^8 for the most frequent item.

For a few attributes, such as `ASOURCE` and `RACE`, even an attacker using the random guessing strategy succeeds more often than may be acceptable. In these cases, higher values of r cannot help limit the adversary's success. These attributes had few possible plaintext values (5 and 6 respectively) and their unencoded distributions were skewed: for example, the most common `ASOURCE` value was about 2^9 times more frequent than the least common value. As we noted above, such guessing attacks are unavoidable in this situation.

One of the strengths of interval-based homophonic encoding is the tuning of parameters it allows: a value of r can be chosen that strikes the right balance between security and efficiency for the intended application. However, users of this scheme should be aware that, in common with any other encryption scheme, it cannot prevent simple guessing attacks. These can be effective for skewed distributions.

7 Related work

In this section, we compare our work to related work—schemes that attempt to hide plaintext frequencies, or prevent frequency analysis. These schemes include order-preserving encryption (OPE) schemes and searchable encryption schemes.

As noted in the introduction, homophonic substitution is a classical cryptographic technique introduced to combat frequency analysis on substitution ciphers (which, after all, is what a DE scheme is). While the idea of applying it in the current domain is not groundbreaking, we present the original analysis required to assess its security in theory and practice. In particular, we did not find our MLE analysis from Section 6.2 in the literature on this topic.

The first OPE scheme [2] uses a kind of homophonic encoding in its construction. Its goal is not necessarily to hide frequencies, but to hide the input's distribution by transforming it to have some target distribution. The paper used the Kolmogorov-Smirnov test to determine whether (i) the input data's distribution was indistinguishable from uniform after flattening, and (ii) the encoded data's distribution was indistinguishable from data with the target distribution (Gaussian, Zipf, or uniform). In their experiments, the data items had 32 bits and encodings had 64 bits. In contrast to [2], our work applies to any type of data, not just numeric, and we focus on DE rather than OPE. Both of our HE schemes can be combined with OPE in an analogous way to our $(\mathsf{HE}, \mathsf{DE})-\mathsf{FSE}$ construction to produce an FSE scheme that is order-preserving. However, OPE schemes suffer from high leakage even without repeated messages, so we have not pursued that direction.

Recent work by Kerschbaum describes a frequency-hiding OPE scheme [16]. The security notion used is indistinguishability under frequency-analysing ordered chosen plaintext attack (IND-FA-OCPA). The adversary is tasked with distinguishing between encryptions of two equal-length sequences of plaintexts, not necessarily distinct, which have at least one randomized order in common (this being a ranking in which ties are allowed to be broken arbitrarily). The IND-FA-OCPA security notion captures the idea that the ciphertext leaks only the randomized order. It does not leak any frequency information, since each message and ciphertext value occurs exactly once. For snapshot attacks against this scheme, see [10]. Roche *et al.* [25] introduced a partial order-preserving encoding scheme that uses the same security notion. This approach is incomparable to ours since we do not require ciphertexts to be distinct. Allowing repetition in turn enables us to achieve more flexible trade-offs between security and performance.

Papadimitriou *et al.*'s splayed additively symmetric homomorphic encryption (SPLASHE) construction [22] hides frequencies while supporting aggregate operations such as `COUNT` and `SUM` by expanding each column into as many columns as there are possible values. Their enhanced SPLASHE construction addresses the attendant storage expansion by assigning individual columns to the “most frequent” values and grouping together the “least frequent” values in one column. To distinguish the less frequent values, a column of deterministically encrypted (DE) values is added. The frequencies of the “least frequent” values in this column are smoothed with a rudimentary padding technique. The threshold separating most frequent and least frequent values is chosen to ensure that there are enough records having their own columns so that their entries in the DE column can be used to equalize the counts of the least frequent values’ DE values. SPLASHE was designed for data analytics and in particular it does not support equality queries or joins. It also suffers from significant data expansion, about 10x for a real-world analytics database.

Another recent construction is a secure order-preserving indexing (OPI) that supports efficient point and range queries while hiding frequencies [18]. OPI expands the plaintext domain to the ciphertext domain by assigning an interval of indices to each plaintext whose size is proportional to its frequency, much like we do with IBHE in Section 4.2. However, there is no formal security analysis nor suggestion about how to choose the size of the ciphertext domain. The schemes we propose have adjustable parameters to attain the desired balance of security and efficiency.

We imagine FSE applied to columns in a database, and there exist other solutions for securely querying an encrypted database. For example, Kamara and Moataz [14] developed a structured encryption scheme for relational databases that supports many types of SQL queries and does not leak any frequency information. However, the storage cost can be very high, and unlike our schemes, it is not a scheme that could be added to an existing SQL database in a legacy-friendly manner; it would entirely replace a database and change how queries are treated.

8 Conclusions and applications

Deterministic encryption has many useful applications, but as recent research has demonstrated, the frequency information it leaks can be devastating to security. Using our approach based on homophonic encoding (HE) lets data owners gain control over how much information their encrypted data leaks when it is at rest. We have provided an empirical evaluation of our approach for moderate parameters, in the spirit of accelerated provable security [11]. We used the same metric as Naveed *et al.* did in their inference attacks on DE [20]: the proportion of items that the attacker successfully recovers in a maximum likelihood attack. FSE can withstand attackers that know the data’s actual distribution, which DE cannot. We showed that our approach rapidly reduces the success rate of such an attacker to that of the trivial guessing strategy (which cannot be prevented by any cryptographic means) as r, the encoding parameter of the IBHE scheme, increases. In passing, we note that our approach can further impede attacks by disguising the number of plaintexts in a column, making it harder to identify which column corresponds to which encrypted attribute.

Encrypting values in database columns to preserve query capabilities is only one application of deterministic encryption. Many OPE scheme are deterministic, while some searchable encryption schemes use deterministically-encrypted per-document keyword tags to find search results. These schemes are then susceptible to frequency analysis attacks. In future work, we plan to explore the application of HE to these areas. In particular, as we have already noted, our HE schemes are compatible with OPE: OPE can simply replace DE in the construction of Section 3.3; our IBHE and BHE schemes do not rely on messages being ordered by frequency, and they work equally well when the messages are in numerical order. Moreover, numerical ordering is preserved by the HE schemes. However, the recent snapshot attack [10] on the FH-OPE scheme of Kerschbaum [16] suggests caution is warranted here.

Relatedly, it would be interesting to determine the effect of HE on the success of pairwise column attacks like those described in [7] (see also [10]). Those attacks were specific to OPE, but it would also be instructive to look at such attacks on DE-encrypted database columns which may be weakly correlated, and assess the impact of applying our HE techniques. Addressing the same issue would be of great interest for indices in searchable encryption, especially in view of the attacks in [6].

Finally, our general definition of FSE is conducive to the development of schemes that can adapt to changing distributions in the underlying data. Relatedly, it is important to assess how the attack prevention capability of our static HE techniques degrades as D changes gradually, to understand how much change can be tolerated.

References

[1] Agency for Healthcare Research and Quality, Rockville, MD. HCUP Nationwide Inpatient Sample (NIS), Healthcare Cost and Utilization Project (HCUP), 2009. `http://www.hcup-us.ahrq.gov/nisoverview.jsp`.

[2] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Order preserving encryption for numeric data. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, SIGMOD '04, pages 563–574, New York, NY, USA, 2004. ACM.

[3] T. Baignères, P. Junod, and S. Vaudenay. How far can we go beyond linear cryptanalysis? In P. J. Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 432–450. Springer, Heidelberg, Dec. 2004.

[4] M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers. Format-preserving encryption. In M. J. Jacobson Jr., V. Rijmen, and R. Safavi-Naini, editors, *SAC 2009*, volume 5867 of *LNCS*, pages 295–312. Springer, Heidelberg, Aug. 2009.

[5] J. Black and P. Rogaway. Ciphers with arbitrary finite domains. In B. Preneel, editor, *CT-RSA 2002*, volume 2271 of *LNCS*, pages 114–130. Springer, Heidelberg, Feb. 2002.

[6] D. Cash, P. Grubbs, J. Perry, and T. Ristenpart. Leakage-abuse attacks against searchable encryption. In I. Ray, N. Li, and C. Kruegel:, editors, *ACM CCS 15*, pages 668–679. ACM Press, Oct. 2015.

[7] F. B. Durak, T. M. DuBuisson, and D. Cash. What else is revealed by order-revealing encryption? In E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, editors, *ACM CCS 16*, pages 1155–1166. ACM Press, Oct. 2016.

[8] P. Grubbs, R. McPherson, M. Naveed, T. Ristenpart, and V. Shmatikov. Breaking web applications built on top of encrypted data. In E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, editors, *ACM CCS 16*, pages 1353–1364. ACM Press, Oct. 2016.

[9] P. Grubbs, T. Ristenpart, and V. Shmatikov. Why your encrypted database is not secure. In *Proceedings of the 16th Workshop on Hot Topics in Operating Systems (HotOS XVI)*, May 2017.

[10] P. Grubbs, K. Sekniqi, V. Bindschaedler, M. Naveed, and T. Ristenpart. Leakage-abuse attacks against order-revealing encryption. In *2017 IEEE Symposium on Security and Privacy*, pages 655–672. IEEE Computer Society Press, May 2017.

[11] V. T. Hoang, T. Krovetz, and P. Rogaway. Robust authenticated-encryption AEZ and the problem that it solves. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 15–44. Springer, Heidelberg, Apr. 2015.

[12] M. S. Islam, M. Kuzu, and M. Kantarcioglu. Access pattern disclosure on searchable encryption: Ramification, attack and mitigation. In *NDSS 2012*. The Internet Society, Feb. 2012.

[13] D. Kahn. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet (2nd edition)*. Scribner, Oct. 1997.

[14] S. Kamara and T. Moataz. SQL on structurally-encrypted databases. Cryptology ePrint Archive, Report 2016/453, 2016. `http://eprint.iacr.org/2016/453`.

[15] G. Kellaris, G. Kollios, K. Nissim, and A. O'Neill. Generic attacks on secure outsourced databases. In E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, editors, *ACM CCS 16*, pages 1329–1340. ACM Press, Oct. 2016.

[16] F. Kerschbaum. Frequency-hiding order-preserving encryption. In I. Ray, N. Li, and C. Kruegel:, editors, *ACM CCS 15*, pages 656–667. ACM Press, Oct. 2015.

[17] M.-S. Lacharité and K. G. Paterson. A note on the optimality of frequency analysis vs. ℓ_p-optimization. Cryptology ePrint Archive, Report 2015/1158, 2015. `http://eprint.iacr.org/2015/1158`.

[18] S. S. Moghadam, G. Gavint, and J. Darmonti. A secure order-preserving indexing scheme for outsourced data. In *2016 IEEE International Carnahan Conference on Security Technology (ICCST)*, pages 1–7, Oct 2016.

[19] B. Morris, P. Rogaway, and T. Stegers. How to encipher messages on a small domain. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 286–302. Springer, Heidelberg, Aug. 2009.

[20] M. Naveed, S. Kamara, and C. V. Wright. Inference attacks on property-preserving encrypted databases. In I. Ray, N. Li, and C. Kruegel:, editors, *ACM CCS 15*, pages 644–655. ACM Press, Oct. 2015.

[21] Oracle. MySQL 5.7 reference manual, 2017. `https://dev.mysql.com/doc/refman/5.7/en/storage-requirements.html`.

[22] A. Papadimitriou, R. Bhagwan, N. Chandran, R. Ramjee, A. Haeberlen, H. Singh, A. Modi, and S. Badrinarayanan. Big data analytics over encrypted datasets with seabed. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 587–602, GA, 2016. USENIX Association.

[23] D. Pouliot and C. V. Wright. The shadow nemesis: Inference attacks on efficiently deployable, efficiently searchable encryption. In E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, editors, *ACM CCS 16*, pages 1341–1352. ACM Press, Oct. 2016.

[24] T. Ristenpart and S. Yilek. The mix-and-cut shuffle: Small-domain encryption secure against N queries. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 392–409. Springer, Heidelberg, Aug. 2013.

[25] D. S. Roche, D. Apon, S. G. Choi, and A. Yerukhimovich. POPE: Partial order preserving encoding. In E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, editors, *ACM CCS 16*, pages 1131–1142. ACM Press, Oct. 2016.

[26] P. Rogaway. Nonce-based symmetric encryption. In B. K. Roy and W. Meier, editors, *FSE 2004*, volume 3017 of *LNCS*, pages 348–359. Springer, Heidelberg, Feb. 2004.

[27] P. Rogaway and T. Shrimpton. Deterministic authenticated-encryption: A provable-security treatment of the key-wrap problem. Cryptology ePrint Archive, Report 2006/221, 2006. `http://eprint.iacr.org/2006/221`.

[28] P. Rogaway and T. Shrimpton. A provable-security treatment of the key-wrap problem. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 373–390. Springer, Heidelberg, May / June 2006.

[29] Y. Zhang, J. Katz, and C. Papamanthou. All your queries are belong to us: The power of file-injection attacks on searchable encryption. In T. Holz and S. Savage, editors, *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, pages 707–720. USENIX Association, 2016.

A Security of an (HE-DE)-FSE scheme

Figure 6 depicts the sequence of games in the proof of smoothness.

Proof. First, consider smoothness of the composed FSE scheme. We prove that $(\mathsf{HE}, \mathsf{DE})$-$\mathsf{FSE}$ is smooth with the given parameters using the sequence of games illustrated in Figure 6. The transitions between successive games are based on indistinguishability and we omit some details of the construction of the corresponding distinguishers for brevity.

Let $\mathcal{A}$ be any SMOOTH adversary for $(\mathsf{HE}, \mathsf{DE})$-$\mathsf{FSE}$ that runs in time at most t, and let Game 0 be the $\mathsf{FSE}{-}\mathsf{SMOOTH}$ game, as in Figure 1. When $b = 0$, the ciphertexts are obtained by sampling messages m_i from $\mathcal{M}$ according to D, encoding them using $\tilde{\mathsf{D}}$ to initialize the state, and then encrypting them. When $b = 1$, the ciphertexts are chosen uniformly at random from a subset of $\mathcal{C}$ of the correct size, the number of FSE homophones of each message.

Let Game 1 be the same as Game 0 except when $b = 0$: the ciphertexts are obtained by first sampling N encodings e_i uniformly at random from the set of HE homophones, and then encrypting them with DE.

Consider the following $(\alpha', \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N)$-$\mathsf{SMOOTH}$ adversary $\mathcal{A}'$ for HE, which will distinguish games 0 and 1. $\mathcal{A}'$ receives $(e_1, \ldots, e_N, \tilde{\mathsf{D}}, \hat{\mathsf{D}})$ and flips a coin $b \in \{0, 1\}$. If $b = 0$, it runs $\mathsf{DE.KeyGen}(\lambda)$ to generate a secret key and encrypts the e_i's with it, resulting in c_i's. If $b = 1$, it runs $\mathsf{HE.Setup}(\lambda, \mathsf{D})$ to generate an initial state s_0^* and samples N c_i's uniformly at random from a subset of $\mathcal{C}$ whose size is $\mathcal{H}^{\mathsf{FSE}}_{\mathsf{s}_0^*}$. It then gives the c_i's, $\tilde{\mathsf{D}}$, and $\hat{\mathsf{D}}$ to $\mathcal{A}$, which returns a bit b'. If $b' = b$, then $\mathcal{A}'$ outputs 1. Otherwise, it outputs 0. By definition, the advantage of $\mathcal{A}'$ is the absolute difference in the probabilities that $\mathcal{A}'$ outputs 1 when its input was real encodings and when its input was uniformly sampled encodings. If $\mathcal{A}'$ received real encodings, then $\mathcal{A}$ is playing game 0. If $\mathcal{A}'$ received uniformly sampled encodings, then $\mathcal{A}$ is playing game 1. Therefore,

$$\begin{aligned}\mathsf{Adv}^{\mathsf{smooth}}_{\mathsf{HE}}(\mathcal{A}', \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N) =& |\mathsf{Adv}^{\mathrm{game0}}_{\mathsf{FSE}}(\mathcal{A}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N) \\ &- \mathsf{Adv}^{\mathrm{game1}}_{\mathsf{FSE}}(\mathcal{A}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N)|\end{aligned}$$

Since HE is $(\alpha_{\mathsf{HE}}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N)$-$\mathsf{SMOOTH}$ for adversaries with unbounded runtime, we have

$$|\mathsf{Adv}^{\mathrm{game0}}_{\mathsf{FSE}}(\mathcal{A}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N) - \mathsf{Adv}^{\mathrm{game1}}_{\mathsf{FSE}}(\mathcal{A}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N)| < \alpha_{\mathsf{HE}}.$$

Next, let Game 2 be the same as Game 1 except when $b = 0$, where the N ciphertexts are chosen from a subset of $\mathcal{C}$ of the right size, with repetitions according to the pattern of repetitions in the randomly selected e_i (but otherwise being sampled without replacement, as in the $b = 1$ case of the $\mathsf{DE{-}PRIV}$ game, cf. Figure 4). We can again build an adversary $\mathcal{A}''$—this time for $\mathsf{DE{-}PRIV}$—that interpolates between games 1 and 2 and has advantage

$$\begin{aligned}&\mathsf{Adv}^{\mathsf{priv}}_{\mathsf{DE}}(\mathcal{A}'', N) \\ =& \left|\mathsf{Adv}^{\mathrm{game1}}_{\mathsf{FSE}}(\mathcal{A}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N) - \mathsf{Adv}^{\mathrm{game2}}_{\mathsf{FSE}}(\mathcal{A}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N)\right|.\end{aligned}$$

$\mathcal{A}''$ flips a coin b and either runs $\mathsf{HE.Setup}(\lambda, \mathsf{D})$ to get an initial state s_0^* , uniformly samples N encoded messages e_i from $\mathcal{H}^{\mathsf{HE}}_{\mathsf{s}_0^*}$, and queries its ENC oracle with the e_i (avoiding repeated queries to ENC when repeated e_i are encountered), or uniformly samples N ciphertexts from a subset of $\mathcal{C}$ having size $|\mathcal{H}^{\mathsf{FSE}}_{\mathsf{s}_0^*}|$. It then runs $\mathcal{A}$ on these N ciphertexts, $\tilde{\mathsf{D}}$, and $\hat{\mathsf{D}}$ and outputs 1 if $\mathcal{A}$'s output b' equals b. Its running time is therefore the time to run $\mathcal{A}$, $t_{\mathsf{HE.Setup}}$, the time to sample N messages (which we assume is less than $N \cdot t_{\mathsf{HE.Encode}}$), and the time it takes to query its oracle (which we assume is instantaneous). Since DE is $(\alpha_{\mathsf{DE}}, t + t_{\mathsf{HE.Setup}} + N \cdot t_{\mathsf{HE.Encode}}, N)$-$\mathsf{PRIV}$,

$$\left|\mathsf{Adv}_{\mathsf{FSE}}^{\text{game1}}(\mathcal{A}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N) - \mathsf{Adv}_{\mathsf{FSE}}^{\text{game2}}(\mathcal{A}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N)\right| < \alpha_{\mathsf{DE}}.$$

Finally, we consider Game 3. In the $b = 0$ case of this game, we now sample the c_i's with replacement from a subset of $\mathcal{C}$ of the right size, no longer relying on the e_i, which were sampled from a set of the same size, to dictate repetitions in the c_i's. It is straightforward to see that the distribution on the c_i's is the same in Game 2 and in Game 3. Hence

$$\left|\mathsf{Adv}_{\mathsf{FSE}}^{\text{game2}}(\mathcal{A}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N) - \mathsf{Adv}_{\mathsf{FSE}}^{\text{game3}}(\mathcal{A}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N)\right| = 0.$$

Finally, since $|\mathcal{H}_{\mathsf{s}_0^*}^{\mathsf{FSE}}| = |\mathcal{H}_{\mathsf{s}_0^*}^{\mathsf{HE}}|$, the $b = 0$ and $b = 1$ cases of Game 3 are identical, so $\mathsf{Adv}_{\mathsf{FSE}}^{\text{game3}}(\mathcal{A}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N) = 0$. We therefore have

$$\begin{aligned}\mathsf{Adv}_{\mathsf{FSE}}^{\mathsf{smooth}}(\mathcal{A}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N) &= \mathsf{Adv}_{\mathsf{FSE}}^{\text{game0}}(\mathcal{A}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N)\\ &< \alpha_{\mathsf{HE}} + \alpha_{\mathsf{DE}}\end{aligned}$$

for any $\mathsf{FSE}-\mathsf{SMOOTH}$ adversary $\mathcal{A}$ running in time at most t.

Next, consider message privacy of the composed scheme. We prove that FSE is $(\alpha_{\mathsf{HE}}+\alpha_{\mathsf{DE}}, t, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N)$-$\mathsf{PRIV}$ by showing that if HE is $(\alpha_{\mathsf{HE}}, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N)$-$\mathsf{HE}-\mathsf{SMOOTH}$ and there is an $(\alpha, t, \tilde{\mathsf{D}}, \hat{\mathsf{D}}, \mathsf{D}, N)$-$\mathsf{PRIV}$ adversary $\mathcal{A}_{\mathsf{FSE}}$ for FSE, then there is also an $(\alpha-\alpha_{\mathsf{HE}}, t+t_{\mathsf{HE.Setup}}+N\cdot(t_{\mathsf{HE.Decode}}+t_{\mathsf{HE.Encode}}), N)$-$\mathsf{PRIV}$ adversary $\mathcal{A}_{\mathsf{DE}}$ for DE.

$\mathcal{A}_{\mathsf{DE}}$ can query its provided encryption oracle ENC_{DE} at most N times (without repetition), while it must simulate encrypting N messages sampled according to D (with repetition) for $\mathcal{A}_{\mathsf{FSE}}$. First, $\mathcal{A}_{\mathsf{DE}}$ initializes the homophonic encoding scheme HE: it runs $\mathsf{HE.Setup}(\lambda, \mathsf{D})$ to generate a state s_0^*. It samples N encodings e_i uniformly at random with replacement from $\mathcal{H}_{\mathsf{s}_0^*}^{\mathsf{HE}}$. It decodes these e_i's to obtain the messages m_i. That is, for $i = 1$ to N, it sets $m_i := \mathsf{HE.Decode}(e_i, \mathsf{s}_0^*)$. Next, it queries ENC_{DE} with each of the distinct encodings e_i to obtain $c_1, \ldots, c_N$. It provides $\mathcal{A}_{\mathsf{FSE}}$ with the distributions $\tilde{\mathsf{D}}$ and $\hat{\mathsf{D}}$, and the N plaintext-ciphertext pairs $((m_1, c_1), \ldots, (m_N, c_N))$. Eventually, $\mathcal{A}_{\mathsf{FSE}}$ outputs a bit b'. $\mathcal{A}_{\mathsf{DE}}$ then outputs the same bit.

Note that $\mathcal{A}_{\mathsf{FSE}}$'s view is exactly the same as in the $\mathsf{FSE}-\mathsf{PRIV}$ game in Figure 2. If ENC_{DE} is operating with $b_{\mathsf{DE}} = 0$ (real ciphertexts), then $\mathcal{A}_{\mathsf{DE}}$ is perfectly simulating the $b = 0$ case for $\mathcal{A}_{\mathsf{FSE}}$ since, by the HE-SMOOTH property, encodings sampled uniformly at random from $\mathcal{H}_{\mathsf{s}_0^*}^{\mathsf{HE}}$ have the same distribution as if they were encodings of messages sampled according to D, with an initial state determined by $\tilde{\mathsf{D}}$.

If ENC_{DE} is operating with $b_{\mathsf{DE}} = 1$ (random bitstrings without replacement), then $\mathcal{A}_{\mathsf{DE}}$ is perfectly simulating the $b = 1$ case for $\mathcal{A}_{\mathsf{FSE}}$. By the $\mathsf{HE}-\mathsf{SMOOTH}$ property, the distribution of encodings of messages sampled according to D is uniform on the set of all homophones $\mathcal{H}_{\mathsf{s}_0^*}^{\mathsf{HE}}$. Since this set of homophones is partitioned into the sets of individual messages' homophones, the distribution on the latter is thus uniform as well. Hence, as required, each message's encoding (and thus its ciphertext) is chosen uniformly at random from a set of the correct size with replacement. Therefore, $\mathcal{A}_{\mathsf{DE}}$'s advantage is at least $\mathcal{A}_{\mathsf{FSE}}$'s advantage less the probability that the HE encodings were distinguishable:

$$\mathsf{Adv}^{\mathsf{priv}}_{\mathsf{DE}}(\mathcal{A}_{\mathsf{DE}}, N) > \alpha - \alpha_{\mathsf{HE}}.$$

The running time of $\mathcal{A}_{\mathsf{DE}}$ is at most the time to run $\mathcal{A}_{\mathsf{FSE}}$, $t_{\mathsf{HE.Setup}}$, sample N values from $\mathcal{H}^{\mathsf{HE}}_{\mathsf{s}^*_0}$ (which we again assume is less than $N \cdot t_{\mathsf{HE.Encode}}$), decode N items, and make at most N queries to its encryption oracle (which we assume is instantaneous), achieving the required bounds. □

B A bound on KL divergence for IBHE

In this Appendix, we prove Lemma 18.

Proof. For ease of notation, suppose $\mathcal{M} = \mathsf{support}(\mathsf{D})$, $\mathcal{E} = \mathcal{H}^{\mathsf{HE}}_{\mathsf{s}} = \{0,1\}^r$, and write $\mathcal{H}^{\mathsf{HE}}$ for $\mathcal{H}^{\mathsf{HE}}_{\mathsf{s}}$. Recall that messages are ordered by increasing frequency, and since $r \geq r_{min-h}$, each message has at least h homophones in $\mathcal{E}$.

Let

$$\delta_i := \lfloor F_{\mathsf{D}}(m_i) \cdot 2^r \rceil - F_{\mathsf{D}}(m_i) \cdot 2^r$$

be a rounding error associated with each message, so $\delta_i \in (-0.5, 0.5]$. For convenience, set $\delta_0 := 0$. Then, we can express the size of a message's homophone set as

$$|\mathcal{H}^{\mathsf{HE}}(m_i)| = f_{\mathsf{D}}(m_i) \cdot 2^r + \delta_i - \delta_{i-1}. \tag{2}$$

In order to apply Theorem 14, the distribution of the encoded data, D_{s}, must already be somewhat close to uniform. This requirement arises when approximating $\log \frac{f_{\mathsf{D}_\mathsf{s}}(e)}{2^{-r}}$ with a second-order MacLaurin series in the analysis of [3] on which Theorem 14 relies. Suppose $e \in \mathcal{H}^{\mathsf{HE}}(m_i)$. By applying eqn. 2 and recalling how D_{s} is defined, we get

$$\frac{f_{\mathsf{D}_\mathsf{s}}(e)}{2^{-r}} = \frac{f_{\mathsf{D}}(m_i) \cdot 2^r}{|\mathcal{H}^{\mathsf{HE}}(m_i)|} = 1 + \frac{\delta_{i-1} - \delta_i}{|\mathcal{H}^{\mathsf{HE}}(m_i)|}.$$

For the approximation to hold, $\frac{\delta_{i-1} - \delta_i}{|\mathcal{H}^{\mathsf{HE}}(m_i)|}$ must be small for all i from 1 to $|\mathcal{M}|$. Since the difference of the rounding errors, $\delta_{i-1} - \delta_i$, could take on any value in the interval $(-1, 1)$, we must instead bound $|\mathcal{H}^{\mathsf{HE}}(m_i)|$ using the fact that $r \geq r_{min-h}$.

We are now able to use the following approximation:

$$
\begin{aligned}
&\mathrm{KL}\left(\mathsf{D_s}, \mathsf{U}_{2^r}\right) \approx \frac{1}{2}\sum_{e\in\mathcal{E}} \frac{(f_{\mathsf{D_s}}(e) - 2^{-r})^2}{2^{-r}} \\
&\approx 2^{r-1}\sum_{e\in\mathcal{E}} (f_{\mathsf{D_s}}(e) - 1/2^r)^2 \\
&\approx 2^{r-1}\sum_{i=1}^{|\mathcal{M}|} |\mathcal{H}^{\mathsf{HE}}(m_i)| \cdot \left(\frac{f_{\mathsf{D}}(m_i)}{|\mathcal{H}^{\mathsf{HE}}(m_i)|} - 1/2^r\right)^2 \\
&\approx 2^{r-1}\sum_{i=1}^{|\mathcal{M}|} \left(\frac{f_{\mathsf{D}}(m_i)^2}{|\mathcal{H}^{\mathsf{HE}}(m_i)|} - \frac{2\cdot f_{\mathsf{D}}(m_i)}{2^r} + \frac{|\mathcal{H}^{\mathsf{HE}}(m_i)|}{2^{2r}}\right) \\
&\approx 2^{r-1}\sum_{i=1}^{|\mathcal{M}|} \left(\frac{f_{\mathsf{D}}(m_i)^2}{|\mathcal{H}^{\mathsf{HE}}(m_i)|}\right) - 1 + \frac{1}{2}.
\end{aligned}
$$

Next, we simplify the sum using eqn. 2:

$$
\begin{aligned}
&\sum_{i=1}^{|\mathcal{M}|} \frac{f_{\mathsf{D}}(m_i)^2}{|\mathcal{H}^{\mathsf{HE}}(m_i)|} = \sum_{i=1}^{|\mathcal{M}|} \frac{\left(|\mathcal{H}^{\mathsf{HE}}(m_i)| - (\delta_i - \delta_{i-1})\right)^2}{2^{2r}\cdot|\mathcal{H}^{\mathsf{HE}}(m_i)|} \\
&= \frac{1}{2^{2r}}\sum_{i=1}^{|\mathcal{M}|} \left(|\mathcal{H}^{\mathsf{HE}}(m_i)| - 2(\delta_i - \delta_{i-1}) + \frac{(\delta_i - \delta_{i-1})^2}{|\mathcal{H}^{\mathsf{HE}}(m_i)|}\right) \\
&= \frac{1}{2^r} + \frac{1}{2^{2r}}\sum_{i=1}^{|\mathcal{M}|} \frac{(\delta_i - \delta_{i-1})^2}{|\mathcal{H}^{\mathsf{HE}}(m_i)|}.
\end{aligned}
$$

where the middle term collapsed to zero by virtue of $\delta_0 = \delta_{|\mathcal{M}|} = 0$. Finally, by noting that $\delta_i \in (-0.5, 0.5]$ guarantees that $(\delta_i - \delta_{i-1})^2 \leq 1$, using the assumption that each message has at least h homophones, and hence that $|\mathcal{M}|$ can be at most $2^r/h$, we get the bound

$$
\sum_{i=1}^{|\mathcal{M}|} \frac{(\delta_i - \delta_{i-1})^2}{|\mathcal{H}^{\mathsf{HE}}(m_i)|} \leq |\mathcal{M}|\frac{1}{h} \leq \frac{2^r}{h^2}.
$$

Combining the equations and inequalities above yields the desired bound:

$$
\mathrm{KL}\left(\mathsf{D_s}, \mathsf{U}_{2^r}\right) \leq \frac{1}{2h^2}.
$$

□

C A bound on KL divergence for BHE

Proof. For ease of notation, suppose $\mathcal{M} = \mathsf{support}(\mathsf{D})$, $\mathcal{E} = \bigcup_{m\in\mathcal{M}} \mathcal{H}_{\mathsf{s}}^{\mathsf{HE}}(m)$, and write $\mathcal{H}^{\mathsf{HE}}$ for $\mathcal{H}_{\mathsf{s}}^{\mathsf{HE}}$. Recall that the number of homophones of $m \in \mathcal{M}$

is its band number, $\lceil 2^l \cdot f_{\mathsf{D}}(m)/f_{\mathsf{D}}(m_{|\mathcal{M}|}) \rceil$, where $m_{|\mathcal{M}|}$ is the most frequent message according to D. Letting

$$\delta_i := |\mathcal{H}^{\mathsf{HE}}(m_i)| - 2^l \cdot f_{\mathsf{D}}(m_i)/f_{\mathsf{D}}(m_{|\mathcal{M}|}),$$

we can write

$$|\mathcal{H}^{\mathsf{HE}}| = \frac{2^l}{f_{\mathsf{D}}(m_{|\mathcal{M}|})} + \sum_{i=1}^{|\mathcal{M}|} \delta_i. \tag{3}$$

By assumption, $|\mathcal{M}| \ll \frac{2^l}{f_{\mathsf{D}}(m_{|\mathcal{M}|})}$, so Theorem 14 applies and we can use the following approximation for the Kullback-Leibler divergence:

$$\begin{aligned}
\mathrm{KL}\left(\mathsf{D}_{\mathsf{s}}, \mathsf{U}_{|\mathcal{H}^{\mathsf{HE}}|}\right) &\approx \frac{1}{2}\sum_{e\in\mathcal{E}} \frac{\left(f_{\mathsf{D}_{\mathsf{s}}}(e) - 1/|\mathcal{H}^{\mathsf{HE}}|\right)^2}{1/|\mathcal{H}^{\mathsf{HE}}|} \\
&\approx \frac{|\mathcal{H}^{\mathsf{HE}}|}{2} \sum_{i=1}^{|\mathcal{M}|} |\mathcal{H}^{\mathsf{HE}}(m_i)| \cdot \left(\frac{f_{\mathsf{D}}(m_i)}{|\mathcal{H}^{\mathsf{HE}}(m_i)|} - \frac{1}{|\mathcal{H}^{\mathsf{HE}}|}\right)^2 \\
&\approx \frac{|\mathcal{H}^{\mathsf{HE}}|}{2} \sum_{i=1}^{|\mathcal{M}|} \left(\frac{f_{\mathsf{D}}(m_i)^2}{|\mathcal{H}^{\mathsf{HE}}(m_i)|} - \frac{2 \cdot f_{\mathsf{D}}(m_i)}{|\mathcal{H}^{\mathsf{HE}}|} + \frac{|\mathcal{H}^{\mathsf{HE}}(m_i)|}{|\mathcal{H}^{\mathsf{HE}}|^2}\right) \\
&\approx \frac{|\mathcal{H}^{\mathsf{HE}}|}{2} \left(\sum_{i=1}^{|\mathcal{M}|} \frac{f_{\mathsf{D}}(m_i)^2}{|\mathcal{H}^{\mathsf{HE}}(m_i)|}\right) - 1 + \frac{1}{2}
\end{aligned}$$

Next, we estimate the sum using the fact that $\delta_i \in [0, 1)$ for $i = 1, \ldots, |\mathcal{M}|$:

$$\begin{aligned}
\sum_{i=1}^{|\mathcal{M}|} \frac{f_{\mathsf{D}}(m)^2}{|\mathcal{H}^{\mathsf{HE}}(m)|} &= \sum_{i=1}^{|\mathcal{M}|} \frac{f_{\mathsf{D}}(m_i)^2}{2^l \cdot f_{\mathsf{D}}(m_i)/f_{\mathsf{D}}(m_{|\mathcal{M}|}) + \delta_i} \\
&\leq \sum_{i=1}^{|\mathcal{M}|} \frac{f_{\mathsf{D}}(m_i)^2}{2^l \cdot f_{\mathsf{D}}(m_i)/f_{\mathsf{D}}(m_{|\mathcal{M}|})} \\
&\leq \frac{f_{\mathsf{D}}(m_{|\mathcal{M}|})}{2^l}.
\end{aligned}$$

Finally, combining this upper bound on the sum with an upper bound on the total number of homophones from Equation 3 yields the desired bound:

$$\begin{aligned}
\mathrm{KL}\left(\mathsf{D}_{\mathsf{s}}, \mathsf{U}_{|\mathcal{H}^{\mathsf{HE}}|}\right) &\leq \frac{\frac{2^l}{f_{\mathsf{D}}(m_{|\mathcal{M}|})} + |\mathcal{M}|}{2} \left(\frac{f_{\mathsf{D}}(m_{|\mathcal{M}|})}{2^l}\right) - \frac{1}{2} \\
&\leq \frac{|\mathcal{M}| \cdot f_{\mathsf{D}}(m_{|\mathcal{M}|})}{2^{l+1}}.
\end{aligned}$$

□

```
Game 0
$b \leftarrow_{\$} \{0,1\}$
if $b = 0$ then
  $\mathsf{sk} \leftarrow \mathsf{DE.KeyGen}(\lambda)$
  $\mathsf{s}_0 \leftarrow \mathsf{HE.Setup}(\lambda, \tilde{\mathsf{D}})$
  $m_1, \dots, m_N \leftarrow_{\mathsf{D}} \mathcal{M}$
  for $i$ in $\{1, \dots, N\}$ do
    $(e_i, \mathsf{s}_i) \leftarrow \mathsf{HE.Encode}(m_i, \mathsf{s}_{i-1})$
    $c_i \leftarrow \mathsf{DE.Encrypt}(\mathsf{sk}, e_i)$
  endfor
else
  $\mathsf{s}_0^* \leftarrow \mathsf{HE.Setup}(\lambda, \mathsf{D})$
  $Y \leftarrow_{\$} \mathcal{C}, |Y| = |\mathcal{H}^{\mathsf{FSE}}_{\mathsf{s}_0^*}|$
  $c_1, \dots, c_N \leftarrow_{\$} Y$
endif
$b' \leftarrow \mathcal{A}(c_1, \dots, c_N, \tilde{\mathsf{D}}, \hat{\mathsf{D}})$
return $(b' = b)$
```

```
Game 1
$b \leftarrow_{\$} \{0,1\}$
if $b = 0$ then
  $\mathsf{sk} \leftarrow \mathsf{DE.KeyGen}(\lambda)$
  $\mathsf{s}_0^* \leftarrow \mathsf{HE.Setup}(\lambda, \mathsf{D})$
  for $i$ in $\{1, \dots, N\}$ do
    $e_i \leftarrow_{\$} \mathcal{H}^{\mathsf{HE}}_{\mathsf{s}_0^*}$
    $c_i \leftarrow \mathsf{DE.Encrypt}(\mathsf{sk}, e_i)$
  endfor
else
  $\mathsf{s}_0^* \leftarrow \mathsf{HE.Setup}(\lambda, \mathsf{D})$
  $Y \leftarrow_{\$} \mathcal{C}, |Y| = |\mathcal{H}^{\mathsf{FSE}}_{\mathsf{s}_0^*}|$
  $c_1, \dots, c_N \leftarrow_{\$} Y$
endif
$b' \leftarrow \mathcal{A}(c_1, \dots, c_N, \tilde{\mathsf{D}}, \hat{\mathsf{D}})$
return $(b' = b)$
```

```
Game 2
$b \leftarrow_{\$} \{0,1\}$
if $b = 0$ then
  $\mathsf{s}_0^* \leftarrow \mathsf{HE.Setup}(\lambda, \mathsf{D})$
  $Y \leftarrow_{\$} \mathcal{C}, |Y| = |\mathcal{H}^{\mathsf{FSE}}_{\mathsf{s}_0^*}|$
  for $i$ in $\{1, \dots, N\}$ do
    $e_i \leftarrow_{\$} \mathcal{H}^{\mathsf{HE}}_{\mathsf{s}_0^*}$
    if $\exists\, j < i : e_i = e_j$ do
      $c_i := c_j$
    else
      $c_i \leftarrow_{\$} Y, Y := Y \setminus \{c_i\}$
    endif
  endfor
else
  $\mathsf{s}_0^* \leftarrow \mathsf{HE.Setup}(\lambda, \mathsf{D})$
  $Y \leftarrow_{\$} \mathcal{C}, |Y| = |\mathcal{H}^{\mathsf{FSE}}_{\mathsf{s}_0^*}|$
  $c_1, \dots, c_N \leftarrow_{\$} Y$
endif
$b' \leftarrow \mathcal{A}(c_1, \dots, c_N, \tilde{\mathsf{D}}, \hat{\mathsf{D}})$
return $(b' = b)$
```

```
Game 3
$b \leftarrow_{\$} \{0,1\}$
if $b = 0$ then
  $\mathsf{s}_0^* \leftarrow \mathsf{HE.Setup}(\lambda, \mathsf{D})$
  $Y \leftarrow_{\$} \mathcal{C}, |Y| = |\mathcal{H}^{\mathsf{FSE}}_{\mathsf{s}_0^*}|$
  $c_1, \dots, c_N \leftarrow_{\$} Y$
else
  $\mathsf{s}_0^* \leftarrow \mathsf{HE.Setup}(\lambda, \mathsf{D})$
  $Y \leftarrow_{\$} \mathcal{C}, |Y| = |\mathcal{H}^{\mathsf{FSE}}_{\mathsf{s}_0^*}|$
  $c_1, \dots, c_N \leftarrow_{\$} Y$
endif
$b' \leftarrow \mathcal{A}(c_1, \dots, c_N, \tilde{\mathsf{D}}, \hat{\mathsf{D}})$
return $(b' = b)$
```

Figure 6: Sequence of games in the proof of smoothness of an (HE, DE)-FSE scheme.

D Targeted attributes

Table 2: The 12 attributes targeted in our experiments.

Attribute	Num. values	Min. bitlength unencoded	r_{min} (IBHE)
Age (AGE)	125	7	20
Admission month (AMONTH)	12	4	4
Admission source (ASOURCE)	5	3	10
Admission type (ATYPE)	6	3	12
Patient died (DIED)	2	1	5
Sex (FEMALE)	2	1	1
Length of stay (LOS)	365	9	23
Major diagnostic category (MDC)	25	5	10
Primary payer (PAY1)	6	3	7
Ethnicity group (RACE)	6	3	7
Disease severity (APRDRG_Severity)	4	2	10
Mortality risk (APRDRG_Risk_Mortality)	4	2	10

E Derivation of the MLE attack

Our analysis relies on the following two assumptions. The first is that a static FSE scheme's Encrypt algorithm outputs each of a message's homophones with equal probability. This property holds for composed FSE schemes arising from both of our static HE constructions. It is reasonable to assume that it would hold for any static FSE scheme since the state is not updated in such schemes and, after all, the goal of a frequency-smoothing scheme is to smooth the distribution to become indistinguishable from uniform. Our second assumption is that the adversary considers only "proper" deterministic decryption functions—its solution cannot map one ciphertext to multiple plaintexts, nor can it assign one plaintext more homophones than it has. This rules out attacks that may otherwise appear to perform well, such as simply guessing that *every* item is the plaintext having the highest frequency in the reference distribution. Such a naive attack could actually perform better than the MLE attack with respect to this metric.

We let $\mathcal{C}'$ denote the collection of N ciphertexts available to the adversary. We let $n(c)$ denote the number of times that ciphertext $c \in \mathcal{C}$ occurs in $\mathcal{C}'$. According to the MLE approach, a most likely decryption θ maximises the likelihood $L(\theta|\mathcal{C}') := \Pr[\mathcal{C}'|\theta]$. Thus we wish to compute

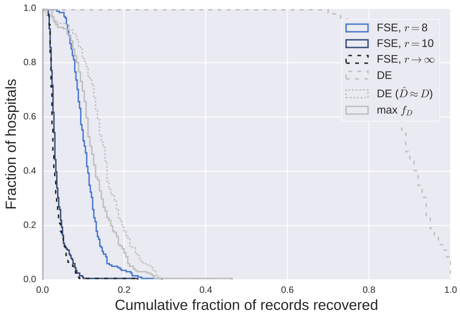
$$\begin{aligned}
\arg\max_{\theta} \Pr\left[\mathcal{C}'|\theta\right] &= \arg\max_{\theta} \prod_{c\in\mathcal{C}} \left(\frac{f_{\mathsf{D}}(\theta(c))}{|\mathcal{H}^{\mathsf{FSE}}(\theta(c))|} \right)^{n(c)} \\
&= \arg\max_{\theta} \prod_{m\in\mathcal{M}} \left(\frac{f_{\mathsf{D}}(m)}{|\mathcal{H}^{\mathsf{FSE}}(m)|} \right)^{\sum_{c\in\theta^{-1}(m)} n(c)} \\
&= \arg\max_{\theta} \sum_{m\in\mathcal{M}} \left(\sum_{c\in\theta^{-1}(m)} n(c) \right) \cdot \log \frac{f_{\mathsf{D}}(m)}{|\mathcal{H}^{\mathsf{FSE}}(m)|}
\end{aligned}$$

where at the last step, we use the fact that maximising a product of terms can be achieved by maximising the sum of the logs of those terms. To maximize this expression, θ should map the most frequently occurring ciphertexts (with largest $n(c)$ values) to the messages with the largest "scaled frequencies" $f_{\mathsf{D}}(m)/|\mathcal{H}^{\mathsf{FSE}}(m)|$. This observation leads directly to the attack given in the main body.

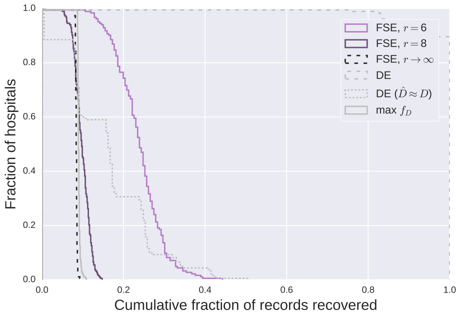
When not all possible ciphertexts appear in the set $\mathcal{C}'$, there is an additional consideration: the sizes of the sets $\theta^{-1}(m)$ no longer need to be equal to the number of homophones of m, $|\mathcal{H}^{\mathsf{FSE}}(m)|$. In this case, we scale the terms $\frac{f_{\mathsf{D}}(m)}{|\mathcal{H}^{\mathsf{FSE}}(m)|}$ in the above analysis and the ensuing attack by an additional factor that is equal to the proportion of all possible ciphertexts that occur in the sample $\mathcal{C}'$.

F Experiment results: FSE-smoothness

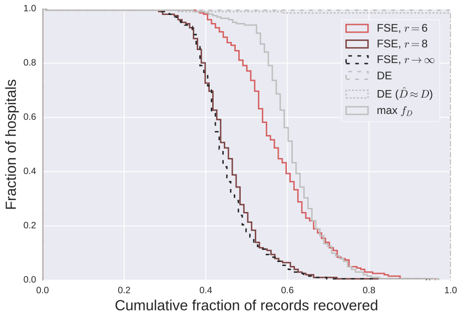
Age



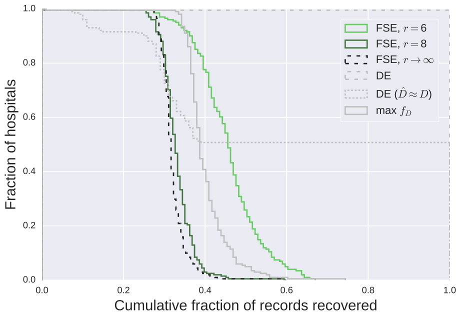
Admission month



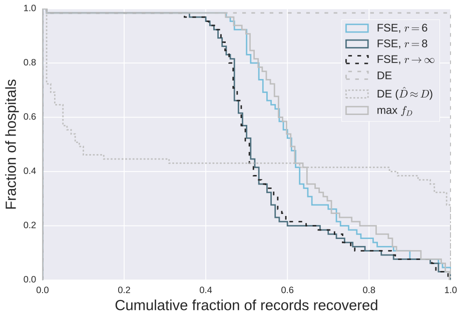
Mortality risk



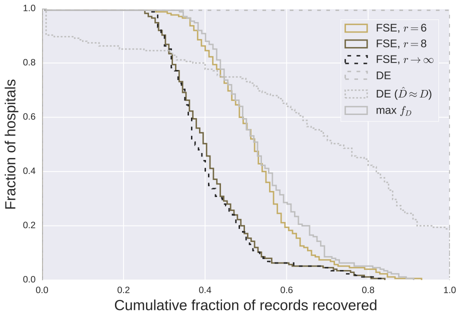
Disease severity



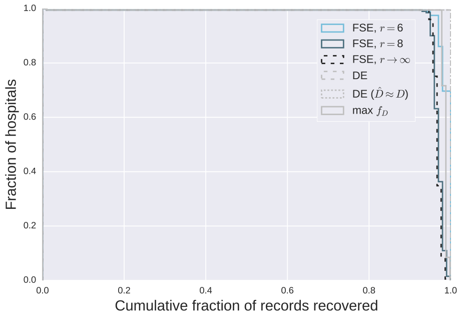
Admission source



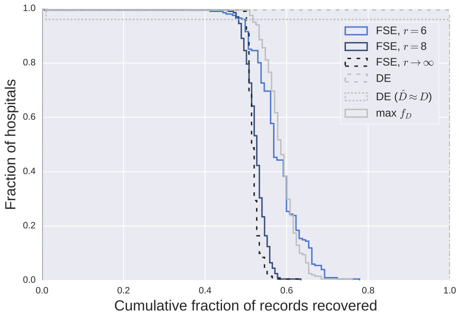
Admission type

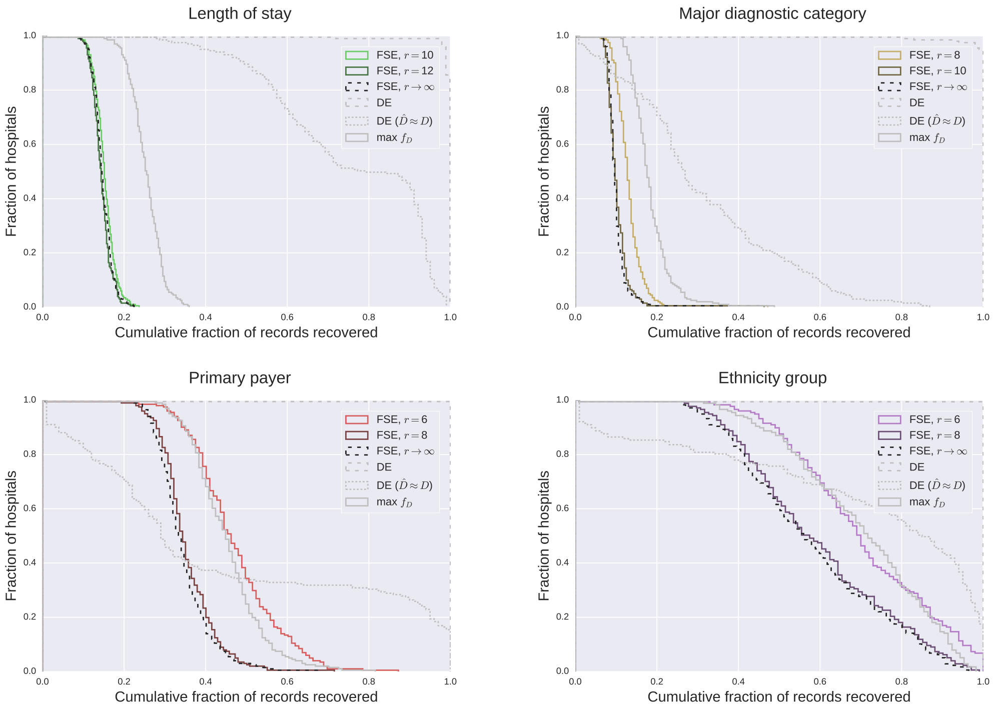


Patient died



Sex




Length of stay
Major diagnostic category
Primary payer
Ethnicity group
Fraction of hospitals
Cumulative fraction of records recovered
0.0
0.2
0.4
0.6
0.8
1.0
FSE, $r = 10$
FSE, $r = 12$
FSE, $r \to \infty$
DE
DE ($\hat{D} \approx D$)
max f_D
FSE, $r = 8$
FSE, $r = 10$
FSE, $r = 6$
FSE, $r = 8$