

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

Marie-Sarah Lacharité and Kenneth G. Paterson

Royal Holloway, University of London

{marie-sarah.lacharite.2015,kenny.paterson}@rhul.ac.uk

Abstract. Statistical analysis of ciphertexts has been recently used to carry out devastating inference attacks on deterministic encryption (Naveed, Kamara, and Wright, CCS 2015), order-preserving/revealing encryption (Grubbs *et al.*, S&P 2017), and searchable encryption (Pouliot and Wright, CCS 2016). At the heart of these inference attacks is classical frequency analysis. In this paper, we propose and evaluate 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 encrypted data, while preserving the ability to efficiently and privately make point queries. 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.

Keywords: database encryption · snapshot attacks · inference attacks · homophonic encoding · frequency-smoothing encryption

1 Introduction

DE for outsourced data. Deterministic Encryption (DE) is an attractive option for encrypting outsourced data 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 its data using an encrypted search term, with the remote data host identifying and returning matches to the user without needing to decrypt them. Similarly, deterministic Order-Preserving/Revealing Encryption (OPE/ORE) allows users to perform efficient range searches on encrypted data, and Symmetric Searchable Encryption (SSE) schemes with deterministically encrypted keywords can use traditional indexing methods to return search results. These schemes have been widely deployed in the industry for protecting data in this way (see, e.g., [FVY⁺17, Sec. I(A)] for a list of commercial solutions).

Frequency analysis and inference attacks. Classical frequency analysis is a powerful attack against deterministically encrypted data. 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 encrypted data, can match frequencies in the encrypted domain with those in the plaintext domain, thus identifying which ciphertext corresponds to which plaintext. This kind of *inference attack*, where statistical techniques are used to infer plaintext information, was used to great

destructive effect in the work of Naveed *et al.* [NKW15]: they correctly inferred large amounts of patient information from DE-encrypted hospital records. Their work and related papers investigating inference attacks [PW16, IKK12, GSB⁺17, BGC⁺17] and leakage-abuse attacks [CGPR15, GMN⁺16, ZKP16, DDC16, KKNO16, LMP18] continue to dent both the industry’s and the research community’s confidence in its ability to adequately protect outsourced data whilst preserving query capabilities.

Countermeasures. Only recently have researchers begun to investigate how to mitigate inference attacks based on frequency analysis. A few proposed OPE/ORE schemes hide frequencies, such as those by Kerschbaum [Ker15] and Boneh *et al.* [BLR⁺15], but they both have limited practicality. In work concurrent to ours, Pouliot, Griffy, and Wright [PGW17] developed the notion of weakly randomized encryption (WRE), in which a small amount of randomness is injected into each ciphertext to prevent frequency analysis. However, their use of statically defined distributions seems to render their most secure construction, WRE with Poisson salt allocation, vulnerable to a form of frequency analysis based on solving knapsack problems, which invalidates the security claim [PGW17, Theorem 4.1]. We discuss this and other related work in greater detail in Section 6.

Homophonic encoding. Given the importance of the problem and the current paucity of solutions, we set out to develop rigorous means of preventing inference attacks on encrypted data, with the particular setting of encrypted databases in mind. Frequency analysis is a venerable attack method, so it is fitting that we were inspired by a technique that is almost as old to counter it: homophonic encoding (HE). The goal of homophonic encoding (or homophonic substitution) is to flatten the frequency distribution of messages by mapping each plaintext to multiple possible *homophones*, with the number of encodings for each plaintext m ideally being proportional to the frequency of m . Then, although homophonically encoded data may still contain repetitions, the homophones occur roughly equally often; frequency information would be of no use to an adversary who has a complete copy of the encoded data.

Homophonic encoding has a long history which is well documented, for example, in [Kah97]. 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 another—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.

HE in the cloud. By combining the power of HE to flatten message distributions and encryption to provide message privacy, we arrive at what we call *frequency-smoothing encryption* (FSE). Using HE leads to encryption schemes that are randomised: we ensure that each message has 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 frequency leakage and increasing query complexity 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 distributions.

Threat model. However, we must immediately issue some important *caveats*. In the current work, we achieve security against only two forms of attack. The first is security in a somewhat randomised generalisation of the standard security notion for DE due to Rogaway and Shrimpton [RS06]. The second is security against inference attacks made by

a snapshot attacker on a per-column basis. Our security proofs and empirical evaluations are respectively focused on these notions. We do not defend against more advanced forms of attack, such as those based on query analysis, as in [CGPR15, GMN⁺16], or attacks based on correlations between columns, as in [DDC16, BGC⁺17]. Concretely, without some kind of query padding or query batching, it will be 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 be roughly proportional to the frequency of m . In addition, Grubbs *et al.* recently pointed out the artificiality of the snapshot attack model [GRS17]. 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 on frequency-smoothing encryption could form the basis of a more complete solution to the problem of preventing inference attacks on encrypted databases.

1.1 Detailed technical contributions

Definitions. 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). FSE is general enough to capture schemes that handle initially unknown or changing message distributions.

Security notions. We provide two security notions for FSE in Section 2. The first, called frequency-smoothing security, prevents frequency analysis attacks by requiring that a collection of FSE ciphertexts be indistinguishable from random data (in a sense to be made precise) even when the underlying plaintext distribution is known. The second, simply called privacy, generalises the symmetric deterministic encryption security notion [RS06]. We carefully motivate our definitional choices in the main body.

Modular construction and proof. 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 via a probabilistic encoding step, flattening the frequency distribution, before encryption with the 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. In Appendix A, we also give a 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.

Specific HE schemes. 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 number of bits of randomness r injected during encryption can be controlled, giving trade-offs between query efficiency and resistance to frequency analysis attacks. Using a novel application of Kullback–Leibler (KL) divergence and based on a framework for optimal distinguishers [BJV04], we show that our HE schemes asymptotically achieve perfect flattening in a statistical sense—even for computationally unbounded adversaries. However, to obtain a bound of typical

cryptographic levels, like 2^{-80} , may require large values of r , which in turn results in high query complexity.

Experimental evaluation. Given this limitation, we use our provable security results as a design guide, and turn to conducting an empirical analysis of the effectiveness of our FSE schemes with moderate values of r . In particular, we evaluate them against attacks which attempt to identify plaintexts with ciphertexts via frequency analysis, in the same way as Naveed *et al.* [NKW15], rather than evaluating them against our formal security notion of FSE-smoothness. This form of attack asks more of the adversary than is required by our formal security definition, so security here offers a weaker guarantee than our formal definition. However, we argue that security in the sense of resisting such attacks is pragmatically useful, given a real-world adversary’s typical aim of recovering actual plaintext. This evaluation is in Section 5.

Generalised frequency analysis attack. 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. [LP15]. 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 [NKW15], 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. We show that FSE is successful in defeating our generalised version of frequency analysis for many attributes, even while maintaining moderate query complexity. The success rate of the MLE adversary is usually quickly reduced to that of a pure guessing strategy for recovering plaintext.

1.2 Terminology and notation

Let \mathbb{D} be any probability distribution on a set of messages \mathcal{M} , and write $f_{\mathbb{D}}(m)$ for the probability mass function (pmf) of a particular message $m \in \mathcal{M}$ according to the distribution \mathbb{D} , so $0 \leq f_{\mathbb{D}}(m) \leq 1$ for all $m \in \mathcal{M}$. The corresponding cumulative density function (cdf) is $F_{\mathbb{D}} : \mathcal{M} \rightarrow [0, 1]$, where $F_{\mathbb{D}}(m_j) = \sum_{i=1}^j f_{\mathbb{D}}(m_i)$ 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 $\text{supp}(\mathbb{D})$ 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 \mathbb{D} , we use $\tilde{\mathbb{D}}$ for the owner’s approximation and $\hat{\mathbb{D}}$ for the adversary’s approximation.

We use \parallel to denote concatenation. $\text{Trunc}(x, n)$ 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.

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 summarizes our notation for these various distributions.

2 Frequency-smoothing encryption (FSE)

Our goal is to design a scheme that outputs ciphertexts whose frequencies are uniform, so even an adversary who knows the underlying plaintext frequencies cannot infer anything about the data. Since such an attacker knows the plaintext distribution, our scheme’s Setup algorithm also accepts as input an estimate $\hat{\mathbb{D}}$ of the messages’ distribution.

Table 1: Overview of our notation for various distributions.

Symbol	Domain	Description
\tilde{D}	\mathcal{M}	owner’s guess of the data’s distribution
\hat{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)

In addition, a *distribution adaptation parameter* Δ indicates how “different” the client’s estimate of the message distribution \tilde{D} may be from the actual message distribution D . The choice of the measure of “difference” will depend on the particular FSE scheme and how it adapts to the message counts it observes. For instance, the parameter Δ may be an upper bound on the Kolmogorov–Smirnov statistic of the two distributions, or their statistical distance. In a sense, this parameter indicates how much uncertainty is associated with the initial estimated distribution \tilde{D} , and thus it indicates how conservatively a dynamic FSE scheme should allocate homophones. Regardless of what measure of difference is used, we assume that $\Delta = 0$ indicates complete confidence that $\tilde{D} = D$, in which case the scheme will be entirely non-adaptive, i.e., static.

Maintaining state is what allows our FSE schemes to handle initially unknown distributions ($\Delta \neq 0$): by updating the state as messages are encrypted, the scheme can allocate more homophones to the more frequently observed messages. Decryption involves accessing the updated state, and therefore the state must always contain enough information to decrypt any message encrypted with an earlier state. The state also makes explicit that encryption requires knowledge of the plaintext distribution, which the client will need to store in practice. Additionally, having a state allows some precomputation on the message distribution to make encryption or decryption faster. Nevertheless, when the precise message distribution is known from the start or the scheme is static ($\Delta = 0$), the state does not need to be updated after running **Setup** and the following definitions simplify 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 statically distributed. If the distribution changes over time, the estimated distribution \tilde{D} given as input to **Setup** 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 quadruple of algorithms $FSE = (\text{Setup}, \text{KeyGen}, \text{Encrypt}, \text{Decrypt})$ such that:

- **Setup** : $\{0, 1\}^* \times \mathcal{D}_{\mathcal{M}} \times \{0, 1\}^* \rightarrow \mathcal{S}$ takes a security parameter $\lambda \in \{0, 1\}^*$, a distribution $\tilde{D} \in \mathcal{D}_{\mathcal{M}}$, and a distribution adaptation parameter $\Delta \in \{0, 1\}^*$ as input and outputs a state $s \in \mathcal{S}$ that includes a description of the distribution \tilde{D} and maybe other information.
- **KeyGen** : $\{0, 1\}^* \rightarrow \mathcal{K}$ takes a security parameter $\lambda \in \{0, 1\}^*$ as input and outputs a secret key $sk \in \mathcal{K}$.
- **Encrypt** : $\mathcal{K} \times \mathcal{M} \times \mathcal{S} \rightarrow \{\mathcal{C} \times \mathcal{S}\} \cup \{\perp\}$ takes a key $sk \in \mathcal{K}$, a message $m \in \mathcal{M}$, and a state $s \in \mathcal{S}$ as input and outputs either a ciphertext $c \in \mathcal{C}$ and an updated state $s' \in \mathcal{S}$ or \perp .

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

Setup, **KeyGen**, and **Encrypt** are randomized algorithms, while **Decrypt** is deterministic. For a particular key sk , call a state s' *attainable* from the state s if $s' = s$ or if there exists a finite sequence of messages $m_1, \dots, m_n \in \mathcal{M}^n$ such that by defining $s_0 := s$ and $(c_i, s_i) \leftarrow \text{Encrypt}(\text{sk}, m_i, s_{i-1})$ for $i = 1, \dots, n$, we get $s_n = s'$ with non-zero probability. A frequency-smoothing scheme is *correct* for a distribution \tilde{D} if for any $s \leftarrow \text{Setup}(\lambda, \tilde{D}, \Delta)$, any $\text{sk} \leftarrow \text{KeyGen}(\lambda)$, any $m \in \text{supp}(\tilde{D})$, and any state s' attainable from s , if $(c, s'') \leftarrow \text{Encrypt}(\text{sk}, m, s')$, then $\text{Decrypt}(\text{sk}, c, s''') = m$ with probability 1 for any s''' attainable from s'' . Less formally, any message encrypted after initialising the state can be decrypted later, even if other messages are encrypted in the meantime, potentially updating the state.

For some fixed λ , \tilde{D} , and Δ , and any key sk output by $\text{KeyGen}(\lambda)$ with non-zero probability, we let $\mathcal{H}_{\text{sk},s}^{\text{FSE}}(m)$ be the set of all possible outputs of $\text{Encrypt}(\text{sk}, m, s')$ such that s_0 is a state output by $\text{Setup}(\lambda, \tilde{D}, \Delta)$ with non-zero probability, s' is attainable from s_0 , and s is attainable from s' . Thus, $\mathcal{H}_{\text{sk},s}^{\text{FSE}}(m)$ is the union of homophone sets of message m for any state that may have come before the state s . We also let $\mathcal{H}_{\text{sk},s}^{\text{FSE}} := \bigcup_{m \in \mathcal{M}} \mathcal{H}_{\text{sk},s}^{\text{FSE}}(m)$ be the set of all possible encryptions (homophones) of any message with key sk for any state that may have come before s . We assume that the sizes of homophone sets are independent of the choice of $\text{sk} \in \mathcal{K}$, so we may write $|\mathcal{H}_s^{\text{FSE}}(m)|$ for $|\mathcal{H}_{\text{sk},s}^{\text{FSE}}(m)|$ and $|\mathcal{H}_s^{\text{FSE}}|$ for $|\mathcal{H}_{\text{sk},s}^{\text{FSE}}|$. Two immediate corollaries of the correctness property are that $\mathcal{H}_{\text{sk},s}^{\text{FSE}}(m) \subseteq \mathcal{H}_{\text{sk},s'}^{\text{FSE}}(m)$ for any state s' attainable from s , and that $\mathcal{H}_{\text{sk},s}^{\text{FSE}}(m_1)$ and $\mathcal{H}_{\text{sk},s}^{\text{FSE}}(m_2)$ are disjoint unless $m_1 = m_2$, in which case $\mathcal{H}_{\text{sk},s}^{\text{FSE}}(m_1) = \mathcal{H}_{\text{sk},s}^{\text{FSE}}(m_2)$.

2.1 Using FSE

To use frequency-smoothing encryption in the intended setting—on outsourced data that is queryable—the set $\mathcal{H}_{\text{sk},s}^{\text{FSE}}(m)$ 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.

FSE does not natively support range queries other than by expanding a range to a set of values and thence to a larger set of homophones. However, the specific constructions for FSE that follow can be adapted to use OPE, in which case range queries can be efficiently supported. We revisit this idea in Section 7.

The state s of an FSE scheme is stored locally at the client, or in a proxy that 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 FSE–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{D} (corresponding to a data owner’s guess of its data’s distribution) to initialize the state and then encrypts messages sampled according to the true distribution D . In the $b = 1$ case, the challenger samples ciphertexts uniformly at random with replacement from a set having the size of the homophone set if the static scheme were used ($\Delta = 0$) with the data’s true distribution D . The adversary receives N ciphertexts, the distribution \tilde{D} that the challenger uses to initialize the state when $b = 0$, its own estimate of the data’s distribution \hat{D} (possibly different from \tilde{D}), and the distribution adaptation parameter Δ . The adversary’s goal is to distinguish these two cases. Informally, if it is able to distinguish the distribution of the N ciphertexts from uniform, then the message distribution must not have been properly smoothed by the FSE scheme.

Game FSE–SMOOTH $_{\text{FSE}}^{\mathcal{A}, \tilde{D}, \hat{D}, D, N, \Delta}(\lambda)$
$b \leftarrow_{\$} \{0, 1\}$ if $b = 0$ then $s_0 \leftarrow \text{FSE.Setup}(\lambda, \tilde{D}, \Delta)$ $\text{sk} \leftarrow \text{FSE.KeyGen}(\lambda)$ $m_1, \dots, m_N \leftarrow_{\mathcal{D}} \mathcal{M}$ for i in $\{1, \dots, N\}$ do $(c_i, s_i) \leftarrow \text{FSE.Encrypt}(\text{sk}, m_i, s_{i-1})$ endfor else $s_0^* \leftarrow \text{FSE.Setup}(\lambda, D, 0)$ $Y \leftarrow_{\$} \mathcal{C}, Y = \mathcal{H}_{s_0^*}^{\text{FSE}} $ $c_1, \dots, c_N \leftarrow_{\$} Y$ endif $b' \leftarrow \mathcal{A}(c_1, \dots, c_N, \tilde{D}, \hat{D}, \Delta)$ return $(b' = b)$

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

Definition 2. Consider the game FSE–SMOOTH in Figure 1. The frequency-smoothing advantage of \mathcal{A} against the FSE scheme FSE is

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

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

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

From the definition of the FSE–SMOOTH game, some necessary conditions are immediately obvious: first, for an FSE scheme to be FSE–SMOOTH for arbitrary \tilde{D} and D , the distribution adaptation parameter would need to be large and so would the total number of homophones—the latter would need to be about the number of samples, N . Therefore, for efficient constructions, it makes sense to consider schemes that are FSE–SMOOTH

for classes of distributions D and \tilde{D} that are “close enough” according to the distribution adaptation parameter Δ .

Second, for a scheme to be FSE–SMOOTH for arbitrarily large N , the size of every message’s homophone set must be proportional to the frequency of the corresponding message according to D . 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 FSE–SMOOTH security notion is comprehensive; it captures the possibility that the attacker has different information (\hat{D}) about the messages’ actual distribution (D) than the data owner used to initialize the state (\tilde{D}). It also captures the possibility that the adversary has information about the data owner’s estimate of the data’s distribution (\tilde{D}). In general, the adversary may not know exactly what distribution the data owner used to initialize the state, but we assume 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 FSE schemes that are provably secure when $D = \tilde{D} = \hat{D}$, while in Section 5, we present results of an empirical analysis of FSE security when $D = \tilde{D} = \hat{D}$ and compare it to security of DE when $\hat{D} = D$ and $\tilde{D} \approx D$.

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 DE schemes [RS06] to our setting. That definition is itself an adaptation of the indistinguishability-from-random-bits (“IND\$”) notion of security for a nonce-based symmetric encryption scheme [Rog04]. It is also similar to the notion of message privacy we use for DE schemes in Section 3.2.

In the detPriv game [RS06], the adversary is tasked with distinguishing real encryptions of messages m of its choice from random bitstrings selected from the ciphertext space. Our FSE–PRIV game diverges from the detPriv game in two related 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. This may seem like a limitation of the adversary’s power, but it reflects exactly the scenario we want to model, one in which encryption depends on the plaintext’s distribution. 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 $|\mathcal{H}_{s_0}^{\text{FSE}}(m)|$.

In the FSE–PRIV game in Figure 2, the challenger either initializes the state using the estimated distribution \tilde{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 in the static scheme ($\Delta = 0$). Given N plaintext-ciphertext pairs, the distributions \hat{D} and \tilde{D} , and the distribution adaptation parameter Δ , the adversary \mathcal{A} must determine how the plaintext-ciphertext pairs were generated.

Game $\text{FSE-PRIV}_{\text{FSE}}^{\mathcal{A}, \tilde{D}, \hat{D}, D, N, \Delta}(\lambda)$
$b \leftarrow_{\$} \{0, 1\}$ $m_1, \dots, m_N \leftarrow_{\mathcal{D}} \mathcal{M}$ if $b = 0$ then $s_0 \leftarrow \text{FSE.Setup}(\lambda, \tilde{D}, \Delta)$ $\text{sk} \leftarrow \text{FSE.KeyGen}(\lambda)$ for i in $\{1, \dots, N\}$ do $(c_i, s_i) \leftarrow \text{FSE.Encrypt}(\text{sk}, m_i, s_{i-1})$ endfor else $s_0^* \leftarrow \text{FSE.Setup}(\lambda, D, 0)$ $Y \leftarrow_{\$} \mathcal{C}, Y = \mathcal{H}_{s_0^*}^{\text{FSE}} $ for i in $\{1, \dots, N\}$ 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}_{s_0^*}^{\text{FSE}}(m_i) $ $Y := Y \setminus Y_{m_i}$ endif $c_i \leftarrow_{\$} Y_{m_i}$ endfor endif $b' \leftarrow \mathcal{A}((m_1, c_1), \dots, (m_N, c_N), \tilde{D}, \hat{D}, \Delta)$ return $(b' = b)$

Figure 2: The privacy game for an FSE scheme.

Definition 4. Consider the message privacy game FSE-PRIV in Figure 2. The message-privacy advantage of \mathcal{A} against the FSE scheme FSE is

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

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

From these definitions, some guidelines arise for creating efficient, secure schemes. First, since homophone set sizes can only increase, the initial homophone set sizes in the $b = 0$ case should be small to leave room to grow the sets corresponding to the most frequent messages. Making a set of homophones too big will only require that some of its members appear with low probability, so the sizes of the final homophone sets in the $b = 0$ case, $|\mathcal{H}_{s_N}^{\text{FSE}}(m)|$, should be roughly equal to the sizes of the homophone sets in the static $b = 1$ case, $|Y_m| = |\mathcal{H}_{s_0^*}^{\text{FSE}}(m)|$.

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 or large enough that no ciphertexts are repeated. 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 an FSE 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 frequencies) with deterministic symmetric-key encryption (to provide privacy). Sections 3.1 and 3.2 present definitions for homophonic encoding and deterministic encryption schemes, while Section 3.3 describes how to compose them to get an FSE scheme. In Appendix A, we describe an alternate construction of an FSE scheme from an HE scheme, a PRF, and an IND $\$$ -CPA secure encryption scheme. By using a conventional IV-based encryption scheme, it becomes possible to skip a potentially expensive decoding step when decrypting FSE ciphertexts.

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 (Setup, Encode, Decode) such that:

- **Setup** : $\{0, 1\}^* \times \mathcal{D}_{\mathcal{M}} \times \{0, 1\}^* \rightarrow \mathcal{S}$ is a probabilistic algorithm that takes a configuration parameter $\lambda \in \{0, 1\}^*$, an estimated distribution \tilde{D} over \mathcal{M} , and a distribution adaptation parameter Δ as input and outputs some state $s \in \mathcal{S}$ that includes a description of the distribution \tilde{D} and any other scheme parameters.
- **Encode** : $\mathcal{M} \times \mathcal{S} \rightarrow \{\mathcal{E} \times \mathcal{S}\} \cup \{\perp\}$ is a probabilistic algorithm that takes a message $m \in \mathcal{M}$ and a state $s \in \mathcal{S}$ as input and outputs either an encoded message $e \in \mathcal{E}$ and an updated state $s' \in \mathcal{S}$, or \perp .
- **Decode** : $\mathcal{E} \times \mathcal{S} \rightarrow \mathcal{M} \cup \{\perp\}$ is a deterministic algorithm that takes an encoded message $e \in \mathcal{E}$ and a state $s \in \mathcal{S}$ 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.

For some fixed λ , \tilde{D} , and Δ , let $\mathcal{H}_s^{\text{HE}}(m)$ be the set of all possible encodings (homophones) of the message $m \in \mathcal{M}$ for any state up to the given state s . Also let $\mathcal{H}_s^{\text{HE}} := \bigcup_{m \in \mathcal{M}} \mathcal{H}_s^{\text{HE}}(m)$. 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 $s' = s$ or there exists some finite sequence of messages $m_1, \dots, m_n \in \mathcal{M}^n$ such that setting $s_0 := s$ and letting $(e_i, s_i) \leftarrow \text{Encode}(m_i, s_{i-1})$ for $i = 1, \dots, n$, then we have $s_n = s'$ with non-zero probability. A homophonic encoding scheme is *correct* for a distribution $\tilde{D} \in \mathcal{D}_{\mathcal{M}}$ if for all states s output by **Setup**($\lambda, \tilde{D}, \Delta$), any message $m \in \text{supp}(\tilde{D})$, and any state s' attainable from s , if $(e, s'') \leftarrow \text{Encode}(m, s')$, then $\text{Decode}(e, s''') = m$ with probability 1 for any s''' attainable from s'' . In particular, the correctness property requires that any two sets of homophones $\mathcal{H}_s^{\text{HE}}(m)$ and $\mathcal{H}_s^{\text{HE}}(m')$ 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 the setting of encrypted databases¹.

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

Game HE–SMOOTH_{HE}^{A, \tilde{D} , \hat{D} , D, N, Δ} (λ)

$b \leftarrow_{\$} \{0, 1\}$

if $b = 0$ **then**

$s_0 \leftarrow \text{HE.Setup}(\lambda, \tilde{D}, \Delta)$

$m_1, \dots, m_N \leftarrow_{\mathcal{D}} \mathcal{M}$

for i **in** $\{1, \dots, N\}$ **do**

$(e_i, s_i) \leftarrow \text{HE.Encode}(m_i, s_{i-1})$

endfor

else

$s_0^* \leftarrow \text{HE.Setup}(\lambda, D, 0)$

$e_1, \dots, e_N \leftarrow_{\$} \mathcal{H}_{s_0^*}^{\text{HE}}$

endif

$b' \leftarrow \mathcal{A}(e_1, \dots, e_N, \tilde{D}, \hat{D}, \Delta)$

return $(b' = b)$

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

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

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

Definition 8. An HE scheme HE is $(\alpha, \tilde{D}, \hat{D}, D, N, \Delta)$ -SMOOTH if for *all* adversaries \mathcal{A} , it holds that $\text{Adv}_{\text{HE}}^{\text{smooth}}(\mathcal{A}, \tilde{D}, \hat{D}, D, N, \Delta) \leq \alpha$.

HE smoothness resembles the Distribution-Transforming Encoder (DTE) security notion from Juels and Ristenpart’s work on honey encryption schemes [JR14]. In that setting, distribution-specific encoders were used to construct encryption schemes that withstand brute-force attacks by yielding plausible plaintexts when decrypting a target ciphertext with incorrect keys. A DTE adversary’s goal is to distinguish between single message-encoding pairs where either the message was sampled according to some given distribution, then encoded, or the encoding was first sampled uniformly at random, then the message obtained by decoding. This notion is tailored to their setting and is less suited to the snapshot inference attacks based on frequency analysis that we are considering. In

¹In 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 instance, by storing a prefix indicating the length of each entry in the column. 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 rather than include a length prefix.

our setting, indistinguishability of a series of samples from one of two distributions is more appropriate than indistinguishability of message-encoding pairs. Nevertheless, the two notions are equivalent in some cases—consider a static HE scheme where the adversary and data owner have perfect distributional knowledge ($D = \tilde{D} = \hat{D}$). Encoded messages in the HE-SMOOTH game can then be decoded by the adversary, yielding a multi-sample version of DTE security. Thus, in this case, HE-SMOOTH security implies DTE security, and when $N = 1$, they are equivalent.

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 strong 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 (KeyGen, Encrypt, Decrypt) with associated sets \mathcal{K} , \mathcal{M} , and \mathcal{C} such that:

- KeyGen : $\{0, 1\}^* \rightarrow \mathcal{K}$ is a probabilistic algorithm that takes a security parameter λ as input and outputs a secret key $\text{sk} \in \mathcal{K}$.
- Encrypt : $\mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$ is a deterministic algorithm that takes a secret key $\text{sk} \in \mathcal{K}$ and a message $m \in \mathcal{M}$ as input, and outputs a ciphertext $c \in \mathcal{C}$.
- Decrypt : $\mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M} \cup \{\perp\}$ is a deterministic algorithm that takes a key $\text{sk} \in \mathcal{K}$ 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 $\text{Decrypt}(\text{sk}, \text{Encrypt}(\text{sk}, m)) = m$ for all $m \in \mathcal{M}$ and all $\text{sk} \in \mathcal{K}$. The security notion we choose to use for DE (Figure 4) is based on indistinguishability from random bits. Such definitions have already been used in the context of nonce-based symmetric encryption [Rog04] and deterministic authenticated encryption (DAE) for key-wrapping [RS06]. 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 (pseudorandom injection) security for DAE [RS06, Section 8], though we dispense with the decryption oracle in that notion.

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

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

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 $\text{Adv}_{\text{DE}}^{\text{priv}}(\mathcal{A}, N) \leq \alpha$.

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

Game $\text{DE-PRIV}_{\text{DE}}^{\mathcal{A}, N}(\lambda)$	$\text{ENC}(m)$
$b \leftarrow_{\$} \{0, 1\}$ $\text{sk} \leftarrow \text{DE.KeyGen}(\lambda)$ $Y \leftarrow_{\$} \mathcal{C}, Y = \mathcal{M} $ $b' \leftarrow \mathcal{A}^{\text{ENC}}$ return $(b' = b)$	if $b = 0$ then $c := \text{DE.Encrypt}(\text{sk}, m)$ 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.

short strings and use a block cipher, or use a small-domain PRP [MRS09, RY13] or a format-preserving encryption scheme [BR02, BRRS09]. For larger domains, a wide-block PRP or an encryption mode such as SIV could be used [RS06].

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

- **Setup** takes a security parameter $\lambda \in \{0, 1\}^*$, a distribution $\mathbf{D} \in \mathcal{D}_{\mathcal{M}}$, and a distribution adaptation parameter $\Delta \in \{0, 1\}^*$ as input. It runs $\text{HE.Setup}(\lambda, \mathbf{D}, \Delta)$ to obtain an initial state s_0 and outputs s_0 .
- **KeyGen** takes a security parameter $\lambda \in \{0, 1\}^*$ as input. It runs $\text{DE.KeyGen}(\lambda)$ to obtain a key $\text{sk} \in \mathcal{K}$ and outputs sk .
- **Encrypt** takes a key $\text{sk} \in \mathcal{K}$, a message $m \in \mathcal{M}$, and a state $s \in \mathcal{S}$ as input. It runs $\text{HE.Encode}(m, s)$ to obtain (e, s') . It then runs $\text{DE.Encrypt}(\text{sk}, e)$ to obtain a ciphertext $c \in \mathcal{C}$. It outputs (c, s') .
- **Decrypt** takes a key $\text{sk} \in \mathcal{K}$, a ciphertext $c \in \mathcal{C}$, and a state $s \in \mathcal{S}$ as input. It runs $\text{DE.Decrypt}(\text{sk}, c)$ to obtain a message $e \in \mathcal{E}$ or \perp . In the former case, it then runs $\text{HE.Decode}(e, s)$ 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.

Theorem 1. *Suppose that HE is an $(\alpha_{\text{HE}}, \tilde{\mathbf{D}}, \hat{\mathbf{D}}, \mathbf{D}, N, \Delta)$ -SMOOTH homophonic encoding scheme on $(\mathcal{M}, \mathcal{E}, \mathcal{S})$ for some $\tilde{\mathbf{D}}, \hat{\mathbf{D}}, \mathbf{D} \in \mathcal{D}_{\mathcal{M}}$ and that DE is an $(\alpha_{\text{DE}}, t + t_{\text{HE.Setup}} + N \cdot (t_{\text{HE.Encode}} + t_{\text{HE.Decode}}), N)$ -PRIV deterministic encryption scheme on $(\mathcal{K}, \mathcal{E}, \mathcal{C})$. Then the FSE scheme $(\text{HE}, \text{DE})\text{-FSE}$ is*

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

Proof. First, consider smoothness of the composed FSE scheme. We prove that (HE, DE)-FSE is smooth with the given parameters using the sequence of games illustrated in Figure 5. The transitions between successive games are based on indistinguishability and we omit some details of the construction of the corresponding distinguishers for brevity.

Game 0	Game 1
<pre> b $\leftarrow_{\\$}$ {0, 1} if $b = 0$ then sk \leftarrow DE.KeyGen(λ) s₀ \leftarrow HE.Setup($\lambda, \tilde{D}, \Delta$) $m_1, \dots, m_N \leftarrow_{\mathcal{D}} \mathcal{M}$ for i in {1, ..., N} do (e_i, s_i) \leftarrow HE.Encode(m_i, s_{i-1}) $c_i :=$ DE.Encrypt(sk, e_i) endfor else s₀[*] \leftarrow HE.Setup($\lambda, D, 0$) $Y \leftarrow_{\\$} \mathcal{C}, Y = \mathcal{H}_{s_0^*}^{\text{FSE}}$ $c_1, \dots, c_N \leftarrow_{\\$} Y$ endif $b' \leftarrow \mathcal{A}(c_1, \dots, c_N, \tilde{D}, \hat{D}, \Delta)$ return ($b' = b$) </pre>	<pre> b $\leftarrow_{\\$}$ {0, 1} if $b = 0$ then sk \leftarrow DE.KeyGen(λ) s₀[*] \leftarrow HE.Setup($\lambda, D, 0$) for i in {1, ..., N} do $e_i \leftarrow_{\\$} \mathcal{H}_{s_0^*}^{\text{HE}}$ $c_i :=$ DE.Encrypt(sk, e_i) endfor else s₀[*] \leftarrow HE.Setup($\lambda, D, 0$) $Y \leftarrow_{\\$} \mathcal{C}, Y = \mathcal{H}_{s_0^*}^{\text{FSE}}$ $c_1, \dots, c_N \leftarrow_{\\$} Y$ endif $b' \leftarrow \mathcal{A}(c_1, \dots, c_N, \tilde{D}, \hat{D}, \Delta)$ return ($b' = b$) </pre>

Figure 5: Sequence of games in the proof of smoothness of an (HE, DE)-FSE scheme (continued on next page).

Let \mathcal{A} be any SMOOTH adversary for (HE, DE)-FSE that runs in time at most t , and let Game 0 be the FSE-SMOOTH game, as in Figure 1. When $b = 0$, the ciphertexts are obtained by sampling messages m_i from \mathcal{M} according to \mathcal{D} , encoding them using \tilde{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 $(\alpha', \tilde{D}, \hat{D}, D, N, \Delta)$ -SMOOTH adversary \mathcal{A}' for HE, which will distinguish games 0 and 1. \mathcal{A}' receives $(e_1, \dots, e_N, \tilde{D}, \hat{D}, \Delta)$ and flips a coin $b \in \{0, 1\}$. If $b = 0$, it runs DE.KeyGen(λ) to generate a secret key and encrypts the e_i 's with it, resulting in c_i 's. If $b = 1$, it runs HE.Setup($\lambda, D, 0$) 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 $|\mathcal{H}_{s_0^*}^{\text{FSE}}|$. It then gives the c_i 's, \tilde{D} , \hat{D} , and Δ 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,

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

Game 2	Game 3
<pre> b \leftarrow $\mathcal{S}\{0, 1\}$ if $b = 0$ then $s_0^* \leftarrow \text{HE.Setup}(\lambda, D, 0)$ $Y \leftarrow \mathcal{C}, Y = \mathcal{H}_{s_0^*}^{\text{FSE}}$ for i in $\{1, \dots, N\}$ do $e_i \leftarrow \mathcal{H}_{s_0^*}^{\text{HE}}$ if $\exists j < i : e_i = e_j$ do $c_i := c_j$ else $c_i \leftarrow \mathcal{S}Y$ $Y := Y \setminus \{c_i\}$ endif endfor else $s_0^* \leftarrow \text{HE.Setup}(\lambda, D, 0)$ $Y \leftarrow \mathcal{C}, Y = \mathcal{H}_{s_0^*}^{\text{FSE}}$ $c_1, \dots, c_N \leftarrow \mathcal{S}Y$ endif $b' \leftarrow \mathcal{A}(c_1, \dots, c_N, \tilde{D}, \hat{D}, \Delta)$ return $(b' = b)$ </pre>	<pre> b \leftarrow $\mathcal{S}\{0, 1\}$ if $b = 0$ then $s_0^* \leftarrow \text{HE.Setup}(\lambda, D, 0)$ $Y \leftarrow \mathcal{C}, Y = \mathcal{H}_{s_0^*}^{\text{FSE}}$ $c_1, \dots, c_N \leftarrow \mathcal{S}Y$ else $s_0^* \leftarrow \text{HE.Setup}(\lambda, D, 0)$ $Y \leftarrow \mathcal{C}, Y = \mathcal{H}_{s_0^*}^{\text{FSE}}$ $c_1, \dots, c_N \leftarrow \mathcal{S}Y$ endif $b' \leftarrow \mathcal{A}(c_1, \dots, c_N, \tilde{D}, \hat{D}, \Delta)$ return $(b' = b)$ </pre>

Figure 5: Sequence of games in the proof of smoothness of an (HE, DE)-FSE scheme (continued from previous page).

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

$$|\text{Adv}_{\text{FSE}}^{\text{game}0}(\mathcal{A}, \tilde{D}, \hat{D}, D, N, \Delta) - \text{Adv}_{\text{FSE}}^{\text{game}1}(\mathcal{A}, \tilde{D}, \hat{D}, D, N, \Delta)| < \alpha_{\text{HE}}.$$

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 DE-PRIV game, cf. Figure 4). We can again build an adversary \mathcal{A}'' —this time for DE-PRIV—that interpolates between games 1 and 2 and has advantage

$$\text{Adv}_{\text{DE}}^{\text{priv}}(\mathcal{A}'', N) = \left| \text{Adv}_{\text{FSE}}^{\text{game}1}(\mathcal{A}, \tilde{D}, \hat{D}, D, N, \Delta) - \text{Adv}_{\text{FSE}}^{\text{game}2}(\mathcal{A}, \tilde{D}, \hat{D}, D, N, \Delta) \right|.$$

\mathcal{A}'' flips a coin b and either runs $\text{HE.Setup}(\lambda, D, 0)$ to get an initial state s_0^* , uniformly samples N encoded messages e_i from $\mathcal{H}_{s_0^*}^{\text{HE}}$, 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 $|\mathcal{H}_{s_0^*}^{\text{FSE}}|$. It then runs \mathcal{A} on these N ciphertexts, \tilde{D} , \hat{D} , and Δ , and outputs 1 if \mathcal{A} 's output b' equals b . Its running time is therefore the time to run \mathcal{A} , $t_{\text{HE.Setup}}$, the time to sample N messages (which we assume is less than $N \cdot t_{\text{HE.Encode}}$), and the time it takes to query its oracle (which we assume is instantaneous). Since DE is $(\alpha_{\text{DE}}, t + t_{\text{HE.Setup}} + N \cdot t_{\text{HE.Encode}}, N)$ -PRIV,

$$\left| \text{Adv}_{\text{FSE}}^{\text{game}1}(\mathcal{A}, \tilde{D}, \hat{D}, D, N, \Delta) - \text{Adv}_{\text{FSE}}^{\text{game}2}(\mathcal{A}, \tilde{D}, \hat{D}, D, N, \Delta) \right| < \alpha_{\text{DE}}.$$

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

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

Finally, since $|\mathcal{H}_{s_0^*}^{\text{FSE}}| = |\mathcal{H}_{s_0^*}^{\text{HE}}|$, the $b = 0$ and $b = 1$ cases of Game 3 are identical, so $\text{Adv}_{\text{FSE}}^{\text{game3}}(\mathcal{A}, \tilde{\mathcal{D}}, \hat{\mathcal{D}}, \mathcal{D}, N, \Delta) = 0$. We therefore have

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

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

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

\mathcal{A}_{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 \mathcal{D} (with repetition) for \mathcal{A}_{FSE} . First, \mathcal{A}_{DE} initializes the homophonic encoding scheme HE: it runs $\text{HE.Setup}(\lambda, \mathcal{D}, 0)$ to generate a state s_0^* . It samples N encodings e_i uniformly at random with replacement from $\mathcal{H}_{s_0^*}^{\text{HE}}$. It decodes these e_i 's to obtain the messages m_i . That is, for $i = 1$ to N , it sets $m_i := \text{HE.Decode}(e_i, s_0^*)$. Next, it queries ENC_{DE} with each of the distinct encodings e_i to obtain c_1, \dots, c_N . It provides \mathcal{A}_{FSE} with the distributions $\tilde{\mathcal{D}}$ and $\hat{\mathcal{D}}$, the distribution adaptation parameter Δ , and the N plaintext-ciphertext pairs $((m_1, c_1), \dots, (m_N, c_N))$. Eventually, \mathcal{A}_{FSE} outputs a bit b' . \mathcal{A}_{DE} then outputs the same bit.

Note that \mathcal{A}_{FSE} 's view is exactly the same as in the FSE–PRIV game in Figure 2. If ENC_{DE} is operating with $b_{\text{DE}} = 0$ (real ciphertexts), then \mathcal{A}_{DE} is perfectly simulating the $b = 0$ case for \mathcal{A}_{FSE} since, by the HE–SMOOTH property, encodings sampled uniformly at random from $\mathcal{H}_{s_0^*}^{\text{HE}}$ have the same distribution as if they were encodings of messages sampled according to \mathcal{D} , with an initial state determined by $\tilde{\mathcal{D}}$.

If ENC_{DE} is operating with $b_{\text{DE}} = 1$ (random bitstrings without replacement), then \mathcal{A}_{DE} is perfectly simulating the $b = 1$ case for \mathcal{A}_{FSE} . By the HE–SMOOTH property, the distribution of encodings of messages sampled according to \mathcal{D} is uniform on the set of all homophones $\mathcal{H}_{s_0^*}^{\text{HE}}$. 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}_{DE} 's advantage is at least \mathcal{A}_{FSE} 's advantage less the probability that the HE encodings were distinguishable:

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

The running time of \mathcal{A}_{DE} is at most the time to run \mathcal{A}_{FSE} , $t_{\text{HE.Setup}}$, sample N values from $\mathcal{H}_{s_0^*}^{\text{HE}}$ (which we again assume is less than $N \cdot t_{\text{HE.Encode}}$), decode N items, and make at most N queries to its encryption oracle (which we assume is instantaneous), achieving the required bounds. \square

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

($\tilde{D} = \hat{D} = D$) and the homophonic encoding scheme is *static* ($\Delta = 0$). We will write $\text{Adv}_{\text{HE}}^{\text{smooth}}(\mathcal{A}, D, N)$ for the adversary’s advantage in this case. 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 HE–SMOOTH adversary’s advantage

When the distribution is public and the HE scheme is static, we can re-interpret the HE–SMOOTH game from Figure 3 in terms of the resulting distribution over the encoded message space \mathcal{E} . Let $D_{\mathbf{s}}$ be this distribution—for a static HE scheme, it depends solely on the initial state \mathbf{s} output by $\text{Setup}(\lambda, D)$. (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, each of its homophones e will have frequency $f_{D_{\mathbf{s}}}(e) = \frac{f_{D_0}(m)}{|\mathcal{H}_{\text{HE}}^{\text{HE}}(m)|}$.

The adversary must distinguish receiving N samples drawn according to $D_{\mathbf{s}}$ and N samples drawn according to the uniform distribution over the set of homophones. We bound an HE–SMOOTH adversary’s distinguishing advantage using a result from Baignères, Junod, and Vaudenay’s statistical framework for analysing distinguishers [BJV04]. It shows 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 , which is defined as

$$\text{KL}(D_0, D_1) := \sum_{m \in \mathcal{M}} f_{D_0}(m) \cdot \log \frac{f_{D_0}(m)}{f_{D_1}(m)}$$

for two distributions D_0 and D_1 having support \mathcal{M} . In particular, when D_1 is the uniform distribution over \mathcal{M} , we can write the KL divergence in terms of D_0 ’s Shannon entropy, $H(D_0)$:

$$\text{KL}(D_0, D_1) = \sum_{m \in \mathcal{M}} f_{D_0}(m) \cdot (\log |\mathcal{M}| + \log f_{D_0}(m)) = \log |\mathcal{M}| - H(D_0).$$

That is, the Kullback–Leibler divergence of D_0 from uniform is the natural log of the support’s size less the Shannon entropy of D_0 in nats.

The following bound follows directly from the result of Baignères, Junod, and Vaudenay that determines an optimal distinguisher’s probability of error [BJV04, Theorem 6].

Theorem 2. *Let HE be a static homophonic encoding scheme with message space \mathcal{M} and encoded message space \mathcal{E} . Let $D \in \mathcal{D}_{\mathcal{M}}$ be a public distribution over \mathcal{M} , and let $D_{\mathbf{s}}$ be the resulting distribution over \mathcal{E} for a state \mathbf{s} output by $\text{HE.Setup}(\lambda, D)$. If $f_{D_{\mathbf{s}}}(e)$ is close to $1/|\mathcal{H}_{\text{HE}}^{\text{HE}}|$ for all encodings $e \in \mathcal{H}_{\text{HE}}^{\text{HE}}$, then, for any HE–SMOOTH adversary \mathcal{A} , and for sufficiently large N ,*

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

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

This theorem applies even to computationally unbounded adversaries. The requirement that $f_{D_s}(e)$ be close to $1/|\mathcal{H}_s^{\text{HE}}|$ for all e is not a restriction; it is necessary for smoothness. (It is inherited from [BJV04, Prop. 5], where it allows a Taylor series expansion to be truncated at the second order with only small error.) Recall that the cdf of the standard normal distribution, Φ , equals 1/2 at 0, so the closer $N \cdot \text{KL}(D_s, U_{|\mathcal{H}_s^{\text{HE}}|})$ is to 0, the smaller is any 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 $\text{KL}(D_s, U_{|\mathcal{H}_s^{\text{HE}}|})$. 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 upper bound on $\text{Adv}_{\text{HE}}^{\text{smooth}}(\mathcal{A}, D, N)$ is given by

$$\text{Adv}_{\text{HE}}^{\text{smooth}}(\mathcal{A}, D, N) \leq \frac{1}{2\sqrt{\pi}} \cdot \sqrt{N \cdot \text{KL}(D_s, U_{|\mathcal{H}_s^{\text{HE}}|})}. \quad (1)$$

This suggests that to make the adversary’s advantage very small, we need $\text{KL}(D_s, U_{|\mathcal{H}_s^{\text{HE}}|}) \ll 1/N$. We now turn to the analysis of two 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 $f_D(m) \cdot 2^r$ 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 in $\text{supp}(D) = \{m_1, m_2, \dots\}$ are numbered by increasing frequency according to D . Now, consider the cumulative distribution F_D . To simplify notation, let $F_D(m_0) := 0$. Then, the homophone set of any message $m_i \in \text{supp}(D)$ is

$$\{ \lfloor 2^r \cdot F_D(m_{i-1}) \rfloor, \dots, \lfloor 2^r \cdot F_D(m_i) \rfloor - 1 \},$$

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

It is clear that the encoding bitlength r must be at least $\log_2 |\text{supp}(D)|$ so each message can have at least one possible encoding. In addition, r must be 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 1. *Let D be a distribution over the message space \mathcal{M} , with messages in $\text{supp}(D) = \{m_1, m_2, \dots\}$ numbered by increasing frequency, and let $h \geq 1$ be a positive integer. Then, when encoded with r -bit IBHE, every message $m \in \text{supp}(D)$ has at least h homophones if and only if $r \geq r_{\min-h}$, where*

$$r_{\min-h} := \left\lceil \max_{1 \leq i \leq |\text{supp}(D)|} \log_2 \frac{i \cdot h - 0.5}{F_D(m_i)} \right\rceil$$

For correctness (i.e., to ensure that no message in the support of D 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 in the support of D to have at least h homophones by noting that messages are ordered according to frequency, so $F_D(m_i) \geq i \cdot f_D(m_1)$.

Corollary 1. *If messages are encoded with r -bit IBHE for some $r \geq \log_2 \frac{h}{f_D(m_1)}$, then every message $m \in \mathcal{M}$ has at least h homophones.*

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

$$\log_2 \frac{h}{f_D(m_1)} \geq \log_2 \frac{i \cdot h - 0.5}{i \cdot f_D(m_1)} \geq \log_2 \frac{i \cdot h - 0.5}{F_D(m_i)}.$$

□

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

Definition 13. The interval-based homophonic encoding (IBHE) scheme with message space $\mathcal{M} \subseteq \{0, 1\}^n$ is defined as follows:

- **Setup** : $(\lambda, D) \mapsto \mathfrak{s}$, computes the maximum r of the minimum encoding length $r_{\min-1}$ and the encoding length $r_{D,\lambda}$ determined by D and λ , and outputs the state $\mathfrak{s} := (r, D)$.
- **Encode** : $(m, \mathfrak{s}) \mapsto e \cup \perp$, chooses an integer e uniformly at random from the set of m 's homophones $\mathcal{H}_\mathfrak{s}^{\text{HE}}(m) := \{ \lfloor 2^r \cdot F_D(m_{i-1}) \rfloor, \dots, \lfloor 2^r \cdot F_D(m_i) \rfloor - 1 \}$, and outputs either the r -bit representation of e , or \perp if $m \notin \text{supp}(D)$.
- **Decode** : $(e, \mathfrak{s}) \mapsto m \cup \perp$, determines the message $m_i \in \text{supp}(D)$ such that $e \in \{F_D(m_{i-1}), \dots, F_D(m_i) - 1\}$, and outputs either $m := m_i$, or \perp if no such m_i exists.

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

$$\{ \lfloor 2^r \cdot F_D(m_{i-1}) \rfloor, \dots, \lfloor 2^r \cdot F_D(m_i) \rfloor - 1 \}$$

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 $4r \cdot |\text{supp}(D)|$ bits.

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

Lemma 1. *Let D be a distribution over \mathcal{M} and suppose that m_1 is the least frequent message in the support of 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 $\mathfrak{s} := (r, D)$. Then,*

$$\text{KL}(D_\mathfrak{s}, U_{2^r}) \leq \frac{1}{2h^2}.$$

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

Let $\delta_i := \lfloor F_D(m_i) \cdot 2^r \rfloor - F_D(m_i) \cdot 2^r$ be a rounding error associated with each message in $\text{supp}(D)$, 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

$$|\mathcal{H}^{\text{HE}}(m_i)| = f_D(m_i) \cdot 2^r + \delta_i - \delta_{i-1}. \quad (2)$$

In order to apply Theorem 2, the distribution of the encoded data, $D_\mathfrak{s}$, must already be somewhat close to uniform. By applying eqn. 2 and recalling how $D_\mathfrak{s}$ is defined, we get

$$\frac{f_{D_\mathfrak{s}}(e)}{2^{-r}} = \frac{f_D(m_i) \cdot 2^r}{|\mathcal{H}^{\text{HE}}(m_i)|} = 1 + \frac{\delta_{i-1} - \delta_i}{|\mathcal{H}^{\text{HE}}(m_i)|}.$$

For the approximation to hold, $\frac{\delta_{i-1}-\delta_i}{|\mathcal{H}^{\text{HE}}(m_i)|}$ 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 $|\mathcal{H}^{\text{HE}}(m_i)|$ using the fact that $r \geq r_{\min} - h$.

Using a Taylor series approximation for $\log(f_{D_s}(e)/2^{-r})$ in the same way as [BJV04, Prop. 5], we obtain

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

Next, we simplify the sum using eqn. 2:

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

where the middle term collapsed to zero by virtue of $\delta_0 = \delta_{|\text{supp}(D)|} = 0$. 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 $|\text{supp}(D)|$ can be at most $2^r/h$, we get the bound

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

Combining the equations and inequalities above yields the desired bound:

$$\text{KL}(D_s, U_{2^r}) \leq \frac{1}{2h^2}.$$

□

Suppose one has a distribution D , N samples, and a given target ϵ for the frequency-smoothing advantage $\text{Adv}_{\text{HE}}^{\text{smooth}}(\mathcal{A}, D, N)$ 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. 1 enables us to derive a minimum bitlength for r to use in the IBHE scheme:

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

Note that to halve the upper bound on an adversary’s advantage ϵ , the minimum encoding length must increase by 1 bit.

A numerical example. Suppose D is such that $f_D(m_1) = 2^{-5}$. 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. 1 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 2^{-10} against IBHE for these parameters.

Variants. We now describe, with practicality in mind, two variants of IBHE, one of which we will use in our evaluation in Section 5.

(Variant 1) Append encodings to messages rather than entirely 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 $\log_2 |\text{supp}(D)|$). Some distributions can yield prohibitively large values of $r_{\min-1}$ if $f_D(m_1)$ is relatively tiny.

The change to how intervals of $\{0, \dots, 2^r - 1\}$ 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 6 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 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, $s := (r, D')$, for all encoding and decoding. The original distribution D does not need to be stored.

4.3 Banded homophonic encoding

We next present a simple homophonic encoding scheme that appends tags to messages rather than replacing them entirely. The tags can have any length $l \geq 1$ and each message has at most 2^l homophones. Let D be some distribution over \mathcal{M} and again suppose that the messages in $\text{supp}(D)$ are numbered according to their frequencies:

$$f_D(m_1) \leq f_D(m_2) \leq \dots \leq f_D(m_{|\text{supp}(D)|}).$$

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 $(0, f_D(m_{|\text{supp}(D)|})]$ into 2^l bands each of width $w := f_D(m_{|\text{supp}(D)|})/2^l$, numbered 1 to 2^l . The messages whose frequencies are in band i , in the interval $((i-1) \cdot w, i \cdot w]$, will each 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 14. The banded homophonic encoding (BHE) scheme with message space $\mathcal{M} \subseteq \{0, 1\}^n$ is defined as follows:

```

Distribution adjustment algorithm
-----
isBigEnough := False
scaleFactor := 1
for i in {1, ..., |supp(D)|} do
  if i = 1 then
    if  $f_D(m_i) < 1/2^{r+1}$  then
       $f_{D'}(m_i) := 1/2^{r+1}$ 
       $scaleFactor := (1 - f_D(m_i))/(1 - f_{D'}(m_i))$ 
    else
       $f_{D'}(m_i) := f_D(m_i)$ 
      // second value could still be too small
  else // i ≥ 2
    if isBigEnough then
       $f_{D'}(m_i) := f_D(m_i)/scaleFactor$ 
    else
      if  $f_D(m_i) \geq 1/2^r \cdot scaleFactor$  then
        isBigEnough := True
         $f_{D'}(m_i) := f_D(m_i)/scaleFactor$ 
      else
         $f_{D'}(m_i) := 1/2^r$ 
         $scaleFactor := (1 - F_D(m_i))/(1 - F_{D'}(m_i))$ 
return  $D'$ 

```

Figure 6: Distribution adjustment algorithm for distribution D and desired encoding length r , with $r \geq \log_2 |\text{supp}(D)|$, and messages numbered by increasing frequency.

- **Setup** : $(\lambda, D) \mapsto s$ computes the tag length l determined by λ and D , the band width $w := f_D(m_{|\text{supp}(D)|})/2^l$, and outputs $s := (l, w, D)$.
- **Encode** : $(m, s) \mapsto m||t \cup \perp$ computes message m 's frequency band, $\mathbf{b} := \lceil f_D(m)/w \rceil$, picks an integer t uniformly at random in $\{0, 1, \dots, \mathbf{b} - 1\}$, and outputs either the $(n + l)$ -bit string $m||t$, where t is represented using l bits, or \perp if $m \notin \text{supp}(D)$.
- **Decode** : $(e, s) \mapsto \text{Trunc}(e, n)$ removes the last l bits of e to recover m .

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 to decode. 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 initially over-sizing l and using a deliberately under-sized set of homophones 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, $|\mathcal{H}_s^{\text{HE}}|$, is not fixed. For Theorem 2 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

$$\delta_i := \lceil 2^l \cdot f_D(m) / f_D(m_{|\text{supp}(D)|}) \rceil - 2^l \cdot f_D(m) / f_D(m_{|\text{supp}(D)|}),$$

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

$$|\mathcal{H}_s^{\text{HE}}| = \frac{2^l}{f_D(m_{|\text{supp}(D)|})} + \sum_{i=1}^{|\text{supp}(D)|} \delta_i.$$

Whereas the total number of homophones was predictable (indeed fixed) for IBHE, here it may vary by as much as $|\text{supp}(D)| - 1$ 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 2, we require $|\text{supp}(D)| \ll \frac{2^l}{f_D(m_{|\text{supp}(D)|})}$. 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.

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

$$\text{KL}(D_s, U_{|\mathcal{H}_s^{\text{HE}}|}) \leq \frac{|\text{supp}(D)| \cdot f_D(m_{|\text{supp}(D)|})}{2^{l+1}}.$$

Proof. For ease of notation, suppose $\mathcal{E} = \bigcup_{m \in \text{supp}(D)} \mathcal{H}_s^{\text{HE}}(m)$, and write \mathcal{H}^{HE} for $\mathcal{H}_s^{\text{HE}}$. Recall that the number of homophones of a message $m \in \text{supp}(D)$ is its band number, $\lceil 2^l \cdot f_D(m) / f_D(m_{|\text{supp}(D)|}) \rceil$, where $m_{|\text{supp}(D)|}$ is the most frequent message according to D . Letting $\delta_i := |\mathcal{H}^{\text{HE}}(m_i)| - 2^l \cdot f_D(m_i) / f_D(m_{|\text{supp}(D)|})$, we can write

$$|\mathcal{H}^{\text{HE}}| = \frac{2^l}{f_D(m_{|\text{supp}(D)|})} + \sum_{i=1}^{|\text{supp}(D)|} \delta_i. \quad (3)$$

By assumption, $|\text{supp}(D)| \ll \frac{2^l}{f_D(m_{|\text{supp}(D)|})}$, so Theorem 2 applies and we can use the following approximation for the Kullback–Leibler divergence:

$$\begin{aligned} \text{KL}(D_s, U_{|\mathcal{H}^{\text{HE}}|}) &\approx \frac{1}{2} \sum_{e \in \mathcal{E}} \frac{(f_{D_s}(e) - 1/|\mathcal{H}^{\text{HE}}|)^2}{1/|\mathcal{H}^{\text{HE}}|} \\ &\approx \frac{|\mathcal{H}^{\text{HE}}|}{2} \sum_{i=1}^{|\text{supp}(D)|} |\mathcal{H}^{\text{HE}}(m_i)| \cdot \left(\frac{f_D(m_i)}{|\mathcal{H}^{\text{HE}}(m_i)|} - \frac{1}{|\mathcal{H}^{\text{HE}}|} \right)^2 \\ &\approx \frac{|\mathcal{H}^{\text{HE}}|}{2} \sum_{i=1}^{|\text{supp}(D)|} \left(\frac{f_D(m_i)^2}{|\mathcal{H}^{\text{HE}}(m_i)|} - \frac{2 \cdot f_D(m_i)}{|\mathcal{H}^{\text{HE}}|} + \frac{|\mathcal{H}^{\text{HE}}(m_i)|}{|\mathcal{H}^{\text{HE}}|^2} \right) \\ &\approx \frac{|\mathcal{H}^{\text{HE}}|}{2} \left(\sum_{i=1}^{|\text{supp}(D)|} \frac{f_D(m_i)^2}{|\mathcal{H}^{\text{HE}}(m_i)|} \right) - 1 + \frac{1}{2} \end{aligned}$$

Next, we estimate the sum using the fact that $\delta_i \in [0, 1)$ for $i = 1, \dots, |\text{supp}(D)|$:

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

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

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

□

Suppose one has a distribution \mathbf{D} , N samples, and a given target ϵ for the frequency-smoothing advantage $\text{Adv}_{\text{HE}}^{\text{smooth}}(\mathcal{A}, \mathbf{D}, N)$ 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

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

Note that since $f_{\mathbf{D}}(m_{|\text{supp}(\mathbf{D})|})$ is the maximum frequency, $f_{\mathbf{D}}(m_{|\text{supp}(\mathbf{D})|}) \geq \frac{1}{|\text{supp}(\mathbf{D})|}$, 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 \mathbf{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 $r \geq 12.7 + \log_2 \frac{1}{f_{\mathbf{D}}(m_1)}$. With banded HE, the minimum *additional* bitlength is $l = 25$.

5 Practical security

We have introduced definitions and general constructions that we proved secure with respect to our expressly defined security notions. However, as we have seen in some numerical examples for our encoding schemes, achieving typical cryptographic security levels for our notion of FSE–SMOOTH security could require large encoding lengths for some distributions, leading to a serious blow-up in query complexity (cf. Sec. 2.1). Given this limitation, we choose to perform an empirical evaluation of the security of FSE against frequency analysis attacks².

In this section, therefore, we adopt a more pragmatic approach, working with moderate encoding lengths and switching to a more practical metric of evaluation, since we already know that we will not attain cryptographic levels of security for arbitrary distributions. The security metric we work with in this section is the number of data items that an attacker can correctly decrypt, which has been used for assessing the effectiveness of inference attacks in the literature [NKW15, GSB⁺17] and closely reflects a real-world adversary's aim of plaintext recovery. This approach is similar to the paradigm of *accelerated provable security*, also called *prove-then-prune* [HKR15]: we designed a scheme and proved its

²Of course, we are also interested in achieving 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.

security based on the security of its primitives, but we relax the primitives for practical use and rely on cryptanalysis to assess security.

We evaluate 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 at the end of that section). Our attacks on FSE are in the public distribution setting, where $\tilde{D} = \hat{D} = D$. This grants the adversary greater power than in the scenario considered by Naveed *et al.* [NKW15], where \hat{D} is only approximately D .

Our aim is to reduce the attacker’s success rate in recovering plaintext to that of a naive guessing attack, which is, in any case, not preventable. We develop a maximum likelihood attack for this setting, and then assess its performance using databases of medical records, the same data to which Naveed *et al.* applied DE and carried out inference attacks [NKW15]. This allows us to compare the security of FSE and of DE, and of FSE to naive guessing attacks.

5.1 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. 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.

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 such that it becomes 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 DB 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 DB . According to the MLE approach, a most likely decryption θ maximises the likelihood $L(\theta|DB) := \Pr[DB|\theta]$. Thus we wish to compute

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

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” $f_D(m)/|\mathcal{H}^{\text{FSE}}(m)|$. This observation leads directly to the following attack.

When not all possible ciphertexts appear in the set DB, the sizes of the sets $\theta^{-1}(m)$ can be strictly less than $|\mathcal{H}^{\text{FSE}}(m)|$. In this case, we scale the number of homophones of each message by the fraction of unique ciphertexts in \mathcal{C} that occur in the sample DB.

So, 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 $|\mathcal{H}_s^{\text{FSE}}(m)|$ for each m in $\text{supp}(D)$, 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. Further, suppose $|\mathcal{H}^{\text{FSE}}| = |\mathcal{C}|$, so that every possible ciphertext appears at least once.

The adversary’s goal is to find the correct many-to-one decryption mapping $\theta : \mathcal{C} \rightarrow \mathcal{M}$. 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. Label the distinct observed ciphertexts so their counts are in decreasing order:

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

Also label the plaintext items in the support of D so their scaled frequencies are in decreasing order:

$$\frac{f_D(m_1)}{|\mathcal{H}_s^{\text{FSE}}(m_1)|} \geq \frac{f_D(m_2)}{|\mathcal{H}_s^{\text{FSE}}(m_2)|} \geq \dots \geq \frac{f_D(m_{|\text{supp}(D)|})}{|\mathcal{H}_s^{\text{FSE}}(m_{|\text{supp}(D)|})|}.$$

Then the attack sets θ so that

$$\begin{aligned} \theta : \{c_1, \dots, c_{|\mathcal{H}_s^{\text{FSE}}(m_1)|}\} &\mapsto m_1, \\ \theta : \{c_{|\mathcal{H}_s^{\text{FSE}}(m_1)|+1}, \dots, c_{|\mathcal{H}_s^{\text{FSE}}(m_1)|+|\mathcal{H}_s^{\text{FSE}}(m_2)|}\} &\mapsto m_2, \end{aligned}$$

and so on, until the $|\mathcal{H}_s^{\text{FSE}}(m_{|\text{supp}(D)|})|$ least frequent ciphertexts are mapped to $m_{|\text{supp}(D)|}$.

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 $|\mathcal{H}_s^{\text{FSE}}(m)| = 1$ for each message m , then this attack reduces to a basic frequency analysis attack of the type used by Naveed, Kamara, and Wright [NKW15], which was shown to be maximum likelihood [LP15]. Thus our attack generalises basic frequency analysis.

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

5.2 Experimental results

We use the aforementioned MLE attack to simulate an attacker attempting to decrypt FSE-encrypted records in a database. We individually attack the 12 columns of each of 200 medical databases (one per hospital). To obtain the distribution 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 [Age09]. The largest hospitals were those with the greatest total number of discharges in that year. The 12 target attributes are listed in Table 2, along with the number of distinct values the attribute can take

and a typical minimum encoding length for the IBHE scheme. (Different per-hospital distributions could result in slightly different r_{min} values.) APRDRG refers to the All Patients Refined Diagnosis Related Groups, a patient classification system.

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 (like in [NKW15]). To simplify our analysis, we ignore all values that were identified as missing, invalid, unavailable, or inconsistent.

Table 2: The 12 attributes targeted in our experiments.

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

Our results are presented in a series of graphs in Figure 7, one for each attribute, and with various encoding lengths r for each attribute. These graphs show complementary cumulative distributions, since we are interested in the number of databases 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. An adversary 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 the binary attribute DIED where there is one very likely plaintext (since most patients survive their hospital visits). Each attribute’s graph in Figure 7 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).

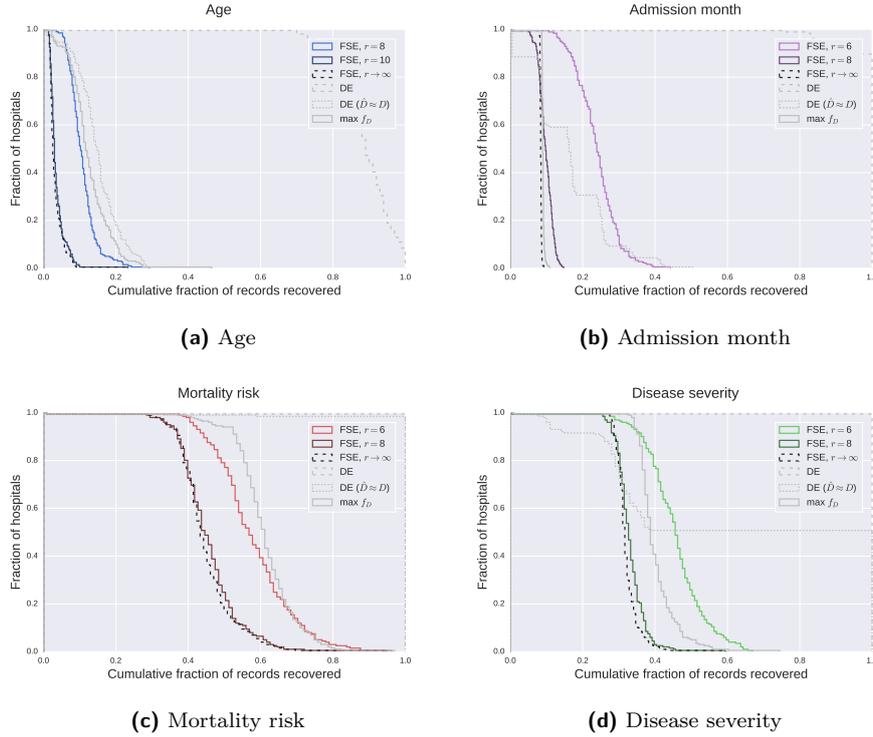


Figure 7: Our experimental results by attribute: complementary cumulative distributions (continued on next two pages).

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 a statistically 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.* individually attacked 200 databases of DE-encrypted medical data from 2009 using aggregated 2004 data for the auxiliary distribution [NKW15]. The power of frequency analysis attacks on DE can be further strengthened by assuming the attacker knows the exact per-database distributions rather than an aggregated distribution. In evaluating DE, we consider both situations, yielding two curves for DE in each graph: one that uses an aggregated distribution ($\hat{D} \approx D$, similar to [NKW15], but from the same year) and the other, a per-database distribution ($\hat{D} = D$). Our experiments attacking FSE always assume that the adversary has exact knowledge of the data’s distribution D , giving it the most power.

For some attributes, frequency analysis on DE even with aggregated auxiliary data recovers nearly *all* records in *all* databases (e.g., APRDRG_Risk_Mortality, DIED, FEMALE), and per-hospital distributions perform even better, recovering nearly 100% of records correctly in every case. And, as can be seen from our graphs in Figure 7, FSE withstands attacks much better than DE in the majority of cases, even when the adversary is given the

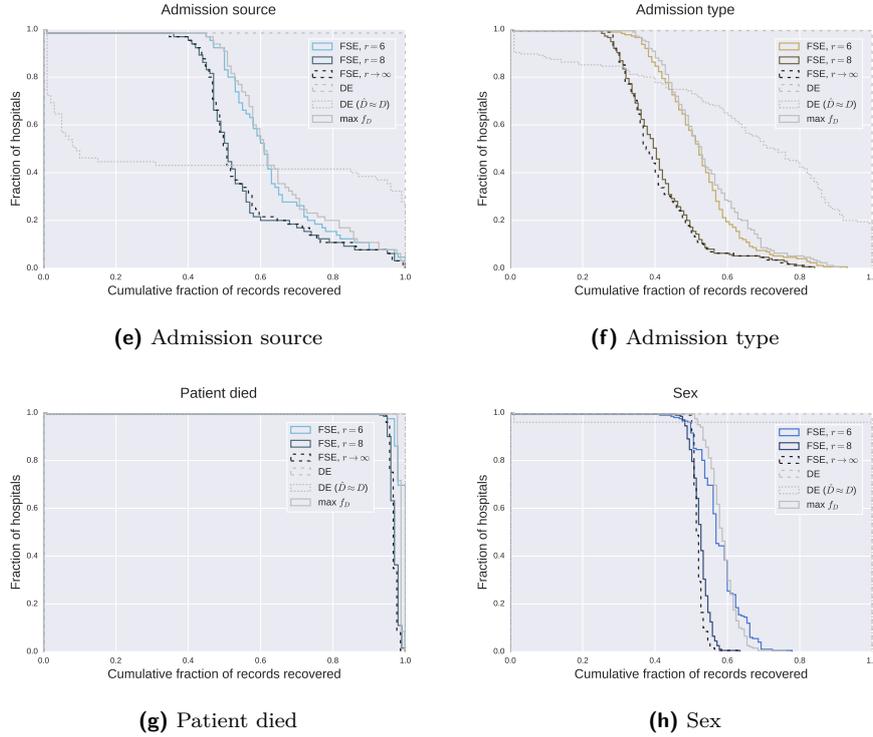


Figure 7: Our experimental results by attribute: complementary cumulative distributions (continued from previous page and continued on next page).

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 earlier, 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 occurs more than once. Given N ciphertext items, our MLE attack assigns approximately $N \cdot f_D(m)$ 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 \mathcal{M} according to \mathbf{D} to determine its guess. The probability that any single ciphertext is assigned the correct plaintext is then $f := \sum_{m \in \mathcal{M}} f_D(m)^2$, 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 $r \rightarrow \infty$ in the graphs. The fraction of records recovered quickly converges to this random guessing strategy, even using encoding lengths much less than r_{min} .

Distribution adjustment algorithm. We use the distribution adjustment algorithm in Figure 6: 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.

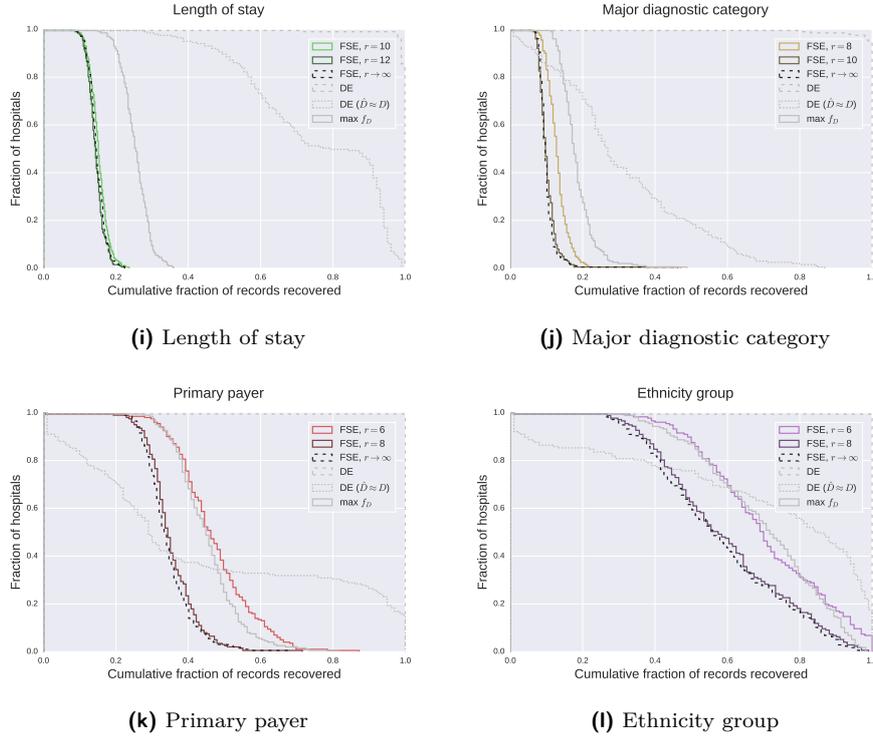


Figure 7: Our experimental results by attribute: complementary cumulative distributions (continued from two previous pages).

The values of r_{min} were typically 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. The parameter r affects query complexity in addition to affecting storage cost: an equality (point) query for one item becomes an equality query for each of its homophones. For large enough encoding lengths r , our results indicate that the statistically optimal MLE attack offers no advantage over guessing—even when the attacker has precise knowledge of the underlying data’s distribution. However, 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 $2^r \cdot f_D(0) \approx 2^7$ 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.

Limitations. 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.

6 Related work

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 5.1 in the literature on this topic.

In concurrent work, Pouliot, Griffy, and Wright [PGW17] developed the notion of weakly randomized encryption (WRE). In such schemes, some randomness is inserted into each ciphertext to prevent frequency analysis, which is also the goal of our frequency-smoothing encryption schemes. Their most secure construction is WRE with Poisson salt allocation, where each plaintext is assigned a number of ciphertexts determined by a Poisson process: for every message m , the Poisson process is run over the interval $(0, f(m)]$, with the number of arrivals determining how many homophones it will have and the inter-arrival times determining what frequency each of them will have. Since the inter-arrival times of the Poisson events are exponentially distributed, the ciphertexts will each have a frequency sampled from an exponential distribution with the same parameter as the Poisson distribution. These ciphertext frequencies are fixed, so without having specified a bound on the number of samples the adversary sees, it is possible to determine their frequencies to arbitrary precision. Then, because the adversary is not computationally bounded, it can exhaustively find groups of ciphertexts for which the sum of their frequencies equals the frequency of one particular plaintext. This enables an unbounded adversary to break the scheme of [PGW17]. It is plausible that a bounded adversary who can solve subset sum problems for certain ranges of parameters would also succeed in breaking the scheme. Of course, this approach requires the auxiliary distribution of plaintext frequencies to be exact.

A few OPE/ORE schemes have properties similar to frequency-smoothing. The first OPE scheme [AKSX04] 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 [AKSX04], 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 (HE, DE)-FSE construction to produce an FSE scheme that is order-preserving.

Kerschbaum [Ker15] presented a frequency-hiding OPE scheme that entirely 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 [GSB⁺17]. 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. Roche *et al.* [RACY16] introduced a partial order-preserving encoding scheme that uses the same security notion. Boneh *et al.*’s ORE scheme [BLR⁺15] is built from multilinear maps and the authors admit it is too inefficient for practical use. These approaches are 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 [PBC⁺16] 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. 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 [MGD16]. 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 [KM16] 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.

7 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 frequency-smoothing encryption (FSE) approach, which is based on homophonic encoding (HE), lets data owners gain control over how much information their encrypted data leaks when it is at rest. Our definitions are generic enough to include schemes that adapt to initially unknown plaintext distributions, and our security notions take into account the number of ciphertexts the adversary sees and its knowledge about the plaintext distribution, which may be different from the data owner’s knowledge about this distribution.

We provided an empirical evaluation of our static (HE,DE)-FSE scheme with interval-based homophonic encoding (IBHE) and moderate encoding lengths r in the case where the data’s distribution is known to both the data owner and adversary. We simulated FSE-encrypting and attacking the same medical records that were DE-encrypted and attacked by Naveed *et al.* [NKW15]. To attack the FSE-encrypted data, we developed a statistically optimal attack that generalises frequency analysis on DE. Then, we were able to directly compare attacking DE and FSE using the same metric: the proportion of items that an adversary successfully recovers. FSE can withstand attacks by adversaries that know the data’s actual distribution, which DE cannot. We showed that our attack on FSE rapidly devolves to having the success rate of a 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, and some

searchable encryption schemes use deterministically encrypted per-document keywords to find search results, which makes them susceptible to inference attacks based on frequency analysis. 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 [GSB⁺17] on the frequency-hiding OPE scheme of Kerschbaum [Ker15] suggests caution is warranted here.

Relatedly, it would be interesting to explore the effect of HE on the success of pairwise column attacks for OPE [DDC16] and on the success of other inference attacks that exploit cross-column correlations [BGC⁺17]. Addressing the same issue would be of great interest for indices in searchable encryption, in particular for inference attacks exploiting word co-occurrence [PW16] or attacks that use subsets of known documents [CGPR15].

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

Acknowledgements. This work was supported by the European Union’s Horizon 2020 research and innovation programme under grant agreement No. H2020-MSCA-ITN-2014-643161 ECRYPT-NET.

References

- [Age09] 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>.
- [AKSX04] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong 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.
- [BGC⁺17] Vincent Bindschaedler, Paul Grubbs, David Cash, Thomas Ristenpart, and Vitaly Shmatikov. The tao of inference in privacy-protected databases. Cryptology ePrint Archive, Report 2017/1078, 2017. <https://eprint.iacr.org/2017/1078>.
- [BJV04] Thomas Baignères, Pascal Junod, and Serge Vaudenay. How far can we go beyond linear cryptanalysis? In Pil Joong Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 432–450. Springer, Heidelberg, December 2004.
- [BLR⁺15] Dan Boneh, Kevin Lewi, Mariana Raykova, Amit Sahai, Mark Zhandry, and Joe Zimmerman. Semantically secure order-revealing encryption: Multi-input functional encryption without obfuscation. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 563–594. Springer, Heidelberg, April 2015.
- [BR02] John Black and Phillip Rogaway. Ciphers with arbitrary finite domains. In Bart Preneel, editor, *CT-RSA 2002*, volume 2271 of *LNCS*, pages 114–130. Springer, Heidelberg, February 2002.
- [BRRS09] Mihir Bellare, Thomas Ristenpart, Phillip Rogaway, and Till Stegers. Format-preserving encryption. In Michael J. Jacobson Jr., Vincent Rijmen, and

- Reihaneh Safavi-Naini, editors, *SAC 2009*, volume 5867 of *LNCS*, pages 295–312. Springer, Heidelberg, August 2009.
- [CGPR15] David Cash, Paul Grubbs, Jason Perry, and Thomas Ristenpart. Leakage-abuse attacks against searchable encryption. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *ACM CCS 15*, pages 668–679. ACM Press, October 2015.
- [DDC16] F. Betül Durak, Thomas M. DuBuisson, and David Cash. What else is revealed by order-revealing encryption? In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 16*, pages 1155–1166. ACM Press, October 2016.
- [FVY⁺17] Benjamin Fuller, Mayank Varia, Arkady Yerukhimovich, Emily Shen, Ariel Hamlin, Vijay Gadepally, Richard Shay, John Darby Mitchell, and Robert K. Cunningham. SoK: Cryptographically protected database search. In *2017 IEEE Symposium on Security and Privacy*, pages 172–191. IEEE Computer Society Press, May 2017.
- [GMN⁺16] Paul Grubbs, Richard McPherson, Muhammad Naveed, Thomas Ristenpart, and Vitaly Shmatikov. Breaking web applications built on top of encrypted data. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 16*, pages 1353–1364. ACM Press, October 2016.
- [GRS17] Paul Grubbs, Thomas Ristenpart, and Vitaly Shmatikov. Why your encrypted database is not secure. In *Proceedings of the 16th Workshop on Hot Topics in Operating Systems (HotOS XVI)*, May 2017.
- [GSB⁺17] Paul Grubbs, Kevin Sekniqi, Vincent Bindschaedler, Muhammad Naveed, and Thomas 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.
- [HKR15] Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway. Robust authenticated-encryption AEZ and the problem that it solves. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 15–44. Springer, Heidelberg, April 2015.
- [IKK12] Mohammad Saiful Islam, Mehmet Kuzu, and Murat Kantarcioglu. Access pattern disclosure on searchable encryption: Ramification, attack and mitigation. In *NDSS 2012*. The Internet Society, February 2012.
- [JR14] Ari Juels and Thomas Ristenpart. Honey encryption: Security beyond the brute-force bound. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 293–310. Springer, Heidelberg, May 2014.
- [Kah97] David Kahn. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet (2nd edition)*. Scribner, Oct. 1997.
- [Ker15] Florian Kerschbaum. Frequency-hiding order-preserving encryption. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *ACM CCS 15*, pages 656–667. ACM Press, October 2015.

- [KKNO16] Georgios Kellaris, George Kollios, Kobbi Nissim, and Adam O’Neill. Generic attacks on secure outsourced databases. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 16*, pages 1329–1340. ACM Press, October 2016.
- [KM16] Seny Kamara and Tarik Moataz. SQL on structurally-encrypted databases. Cryptology ePrint Archive, Report 2016/453, 2016. <http://eprint.iacr.org/2016/453>.
- [LMP18] Marie-Sarah Lacharité, Brice Minaud, and Kenneth G. Paterson. Improved reconstruction attacks on encrypted data using range query leakage. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 1–18. IEEE Computer Society Press, 2018.
- [LP15] Marie-Sarah Lacharité and Kenneth 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>.
- [MGD16] 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.
- [MRS09] Ben Morris, Phillip Rogaway, and Till Stegers. How to encipher messages on a small domain. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 286–302. Springer, Heidelberg, August 2009.
- [NKW15] Muhammad Naveed, Seny Kamara, and Charles V. Wright. Inference attacks on property-preserving encrypted databases. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *ACM CCS 15*, pages 644–655. ACM Press, October 2015.
- [PBC⁺16] Antonis Papadimitriou, Ranjita Bhagwan, Nishanth Chandran, Ramachandran Ramjee, Andreas Haeberlen, Harmeet Singh, Abhishek Modi, and Saikrishna 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.
- [PGW17] David Pouliot, Scott Griffy, and Charles V. Wright. The strength of weak randomization: Efficiently searchable encryption with minimal leakage. Cryptology ePrint Archive, Report 2017/1098, 2017. <https://eprint.iacr.org/2017/1098>.
- [PW16] David Pouliot and Charles V. Wright. The shadow nemesis: Inference attacks on efficiently deployable, efficiently searchable encryption. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 16*, pages 1341–1352. ACM Press, October 2016.
- [RACY16] Daniel S. Roche, Daniel Apon, Seung Geol Choi, and Arkady Yerukhimovich. POPE: Partial order preserving encoding. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 16*, pages 1131–1142. ACM Press, October 2016.
- [Rog04] Phillip Rogaway. Nonce-based symmetric encryption. In Bimal K. Roy and Willi Meier, editors, *FSE 2004*, volume 3017 of *LNCS*, pages 348–359. Springer, Heidelberg, February 2004.

- [RS06] Phillip Rogaway and Thomas Shrimpton. A provable-security treatment of the key-wrap problem. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 373–390. Springer, Heidelberg, May / June 2006.
- [RY13] Thomas Ristenpart and Scott Yilek. The mix-and-cut shuffle: Small-domain encryption secure against N queries. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 392–409. Springer, Heidelberg, August 2013.
- [ZKP16] Yupeng Zhang, Jonathan Katz, and Charalampos Papamanthou. All your queries are belong to us: The power of file-injection attacks on searchable encryption. In Thorsten Holz and Stefan Savage, editors, *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10–12, 2016.*, pages 707–720. USENIX Association, 2016.

A 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 could be more efficient. In this appendix, we describe how to build an FSE scheme of this type, from any HE scheme, a PRF, and a conventional IV-based IND\$-CPA encryption scheme. This approach is somewhat inspired by the synthetic IV (SIV) construction of Rogaway and Shrimpton [RS06]. We modify SIV by using a homophonic encoding of the message instead of the message itself to generate the IVs.

Definition 15. Let $\text{HE} = (\text{Setup}, \text{Encode}, \text{Decode})$ be a stateful homophonic encoding scheme with message space \mathcal{M} and encoded message space \mathcal{E} . Let $\text{CIV} = (\text{KeyGen}, \text{Encrypt}, \text{Decrypt})$ be a conventional IV-based encryption scheme, as defined in [RS06], 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* (HE, CIV)-FSE scheme is defined as follows.

- **Setup** takes a security parameter $\lambda \in \{0, 1\}^*$, a distribution $D \in \mathcal{D}_{\mathcal{M}}$, and a distribution adaptation parameter $\Delta \in \{0, 1\}^*$ as input. It runs $\text{HE.Setup}(\lambda, D, \Delta)$ to obtain an initial state s_0 , which it outputs.
- **KeyGen** takes a security parameter $\lambda \in \{0, 1\}^*$ as input. It runs $\text{CIV.KeyGen}(\lambda)$ to obtain a key $sk_1 \in \mathcal{K}_1$ and selects $sk_2 \leftarrow_s \mathcal{K}_2$. It outputs (sk_1, sk_2) .
- **Encrypt** takes keys $(sk_1, sk_2) \in \mathcal{K}_1 \times \mathcal{K}_2$, a message $m \in \mathcal{M}$, and a state $s \in \mathcal{S}$ as input. It runs $\text{HE.Encode}(m, s)$ to obtain either \perp , in which case it also returns \perp , or (e, s') . It then computes $\text{PRF}(sk_2, e)$ to get $iv \in \mathcal{IV}$. Lastly, it runs $\text{CIV.Encrypt}(sk_1, m; iv)$ to obtain a ciphertext $c \in \mathcal{C}$. It sets $\hat{c} = iv \| c$ and outputs (\hat{c}, s') .
- **Decrypt** takes keys $(sk_1, sk_2) \in \mathcal{K}_1 \times \mathcal{K}_2$ and a ciphertext \hat{c} as input. It parses \hat{c} as $iv \| c \in \mathcal{IV} \times \mathcal{C}$ or returns \perp if this is not possible. It runs $\text{CIV.Decrypt}(sk_1, c; iv)$ to obtain a message m , and returns m .

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

We omit a detailed security analysis of this scheme. Satisfying FSE–PRIV follows from the IND\$-CPA security of CIV. The use of a PRF to generate the IVs from encodings e produces IVs that are indistinguishable from random, up to repetitions induced by the encoding scheme, with such encodings arising only from message repetitions and therefore

resulting in identical ciphertexts $\hat{c} = iv||c$. Satisfying FSE-SMOOTH, on the other hand, follows from the HE-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 deterministic authenticated encryption (DAE) scheme, in the sense introduced in [RS06]. The idea is to set the header for the DAE scheme to be $\text{PRF}(\text{sk}_2, e)$ where e is output by HE.Encode as in the above construction. We note, however, that the integrity properties enjoyed by DAE are overkill for security in our snapshot attacker model.