Hiding the Lengths of Encrypted Messages via Gaussian Padding

Jean Paul Degabriele
CNS, Technische Universität Darmstadt

ABSTRACT

Secure network protocols like TLS, QUIC, SSH and IPsec allow for additional padding to be used during encryption in order to hide message lengths. While it is impossible to conceal message lengths completely, without drastically degrading efficiency, such mechanisms aim at causing as much frustration as possible to the prospective attacker. However, none of the protocol specifications provide any guidance on how to select the length of this padding. Several works have highlighted how the leakage of message lengths can be exploited in attacks, but the converse problem of how to best defend against such attacks remains relatively understudied. We make this the focus of our work and present a formal treatment of length hiding security in a general setting. Prior work by Tezcan and Vaudenay suggested that sampling the padding length uniformly at random already achieves the best possible security. However we show that this is only true in the limited setting where only a single ciphertext is available to the adversary. If multiple ciphertexts are available to the adversary, then sampling the padding length according to a Gaussian distribution yields quantifiably better security for the same overhead. In fact, in this setting, uniformly random padding turns out to be among the worst possible choices. We confirm experimentally the superior performance of Gaussian padding over uniform padding in the context of the CRIME/BREACH attack.

CCS CONCEPTS

• **Security and privacy → Mathematical foundations of cryptography**; **Security protocols**.

KEYWORDS

Length Hiding; Random Padding; Gaussian Padding, Cover Difference, CRIME, BREACH

ACM Reference Format:
Jean Paul Degabriele. 2021. Hiding the Lengths of Encrypted Messages via Gaussian Padding. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21), November 15–19, 2021, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3460120.3484590

1 INTRODUCTION

Conventional wisdom within the cryptographic community has it that encryption cannot conceal message lengths. Information-theoretic formulations of this statement can be found in the work of Chor and Kushilevitz [12], and that of Phan and Vaudenay [37]. It is also reflected in almost all security definitions for encryption, which require that message pairs queried to a left-or-right oracle always be of the same length, or that the encryption scheme have a fixed expansion, i.e., it produces ciphertexts that are a fixed number of bits longer than the corresponding messages. In effect, hiding message lengths is considered to be a lost cause within the theoretical community and typically kept out of the picture, i.e., the security model.

Yet the leakage of message sizes is very problematic in practice and is in fact one of the common avenues for breaking real-world systems [15, 20, 21, 25, 26, 29, 35, 38, 42, 45, 51]. Accordingly, secure network protocols like TLS [48], SSH [52], QUIC [49], and IPsec [27] include length-padding mechanisms in order to hide message lengths and mitigate against this type of attacks. Clearly, any practical length-padding scheme cannot provide an absolute guarantee of protection since it cannot reduce an adversary's success probability to a negligible value. Instead, the goal here is to maximise the adversary's frustration by slowing her down, causing her to consume as many resources as possible, and limiting the damage that she may cause. Moreover, length-padding could be one of a set of countermeasures that, in combination, succeed in thwarting an attack.

Note that a similar situation arises in the realm of Differential Privacy [16], where in order to retain an acceptable level of accuracy in the statistical data, we have to content ourselves with limiting the adversary's advantage only to a moderately small value rather than a negligible one [17]. Nevertheless, Differential Privacy has proven to be an effective and pragmatic solution for protecting the privacy of individuals within a dataset. Accordingly, we think there is scope in studying length padding and how to maximise its efficacy. Length padding incurs a costly overhead in bandwidth, and it is thus essential to obtain the best possible security for a given amount of overhead and to be able to quantify its benefits. However, while the above protocol specifications ensure that implementations can handle length padding correctly, they provide no guidance on how to determine its length. Indeed an excerpt from the TLS 1.3 specification (RFC 8446) reads as follows:

"Selecting a padding policy that suggests when and how much to pad is a complex topic and is beyond the scope of this specification. If the application-layer protocol on top of TLS has its own padding, it may be preferable to pad Application Data TLS records within the application layer. Padding for encrypted Handshake or Alert records must still be handled at the TLS layer, though. Later documents may define padding selection algorithms or define a padding policy request mechanism through TLS extensions or some other means."

Consequently, when random-sized padding is used, the favoured approach is typically the most direct one, i.e., padding of uniformly distributed length, which we refer to as *uniform random padding*.

Guidance on length padding is also sparse within the academic body of literature. To the best of our knowledge, the only formal

CCS '21, November 15–19, 2021, Virtual Event, Republic of Korea

ACM ISBN 978-1-4503-8454-4/21/11…$15.00
https://doi.org/10.1145/3460120.3484590

security treatments of length padding are that of Paterson, Ristenpart and Shrimpton [36], and that of Tezcan and Vaudenay [47], henceforth referred to as PRS11 and TV11, respectively. However, while they consider similar settings, they ultimately focus on fairly different aspects of length padding. In particular, PRS11 studies how length padding may be circumvented at the cryptographic level due to information leaking from the decryption algorithm. Examples of such attacks were described in [36] in the context of TLS, and earlier in [15] for the case of IPsec. On the other hand, TV11 focuses on quantifying the security provided by uniform random padding. Their main results show that the associated security bound degrades linearly with the average padding length, but more interestingly, they show that with respect to their security model, uniform random padding is nearly optimal.

1.1 Our Contribution

In this work, we take a new look at length-hiding security. We build and improve on prior work to obtain a more comprehensive security model and then reduce it, under standard assumptions about the encryption scheme, to a simplified and purely information-theoretic model for evaluating the efficacy of a padding distribution. We then derive an inequality expressing the adversary's multisample advantage as a function of its single-sample advantage and the number of samples. A key insight surfaced by this relation is that the single-sample advantage consists of two components that grow at different rates as the number of samples increases. This insight leads us to the Laplace and Gaussian distributions as superior alternatives to uniform padding. We further confirm this experimentally, both with respect to our security model and the CRIME/BREACH attack. Specifically, in the latter case, we show that at an average overhead of 200 bytes per message switching from uniform to Gaussian padding raises the required number of samples/messages from $76,883$ to $7,680,000$ while simultaneously lowering the success probability from 1.0 to 0.0026. We discuss each of these contributions in more detail below.

Security Definition and Composition. When deciding on the best way to sample length padding, the analysis of TV11 suggests that there is not much to ponder about as the simplest solution (uniform padding) already yields the best performance. However, the security model in which TV11 derived their result was rather restricted. Most notably, it only allowed the adversary to observe a single ciphertext. A well-known technique for weakening the effect of random padding is to 'average out' its effect over multiple queries, an aspect that is not captured in the security model of TV11.

Motivated by this limitation and the work of PRS11 we propose a unified security model that combines the best of both. Our security definition allows us to evaluate the efficacy of different length padding schemes under multiple encryptions while at the same time capturing possibly bad interactions with the cryptographic component. Examples of such bad interactions were exposed against IPsec in [15] and against TLS in [36]. To addresses this, we prove a composition theorem showing that the common pad-then-encrypt strategy is sound, provided that the encryption satisfies a mild security requirement called channel simulatability. In addition, the composition theorem also serves to reduce our comprehensive length-hiding security model to a simpler and purely information-theoretic one that focuses solely on the padding scheme.

Cover Difference and Multisample Distinguisher. We introduce a new simple statistical measure which we call *cover difference*. Intuitively this captures an adversary's ability to distinguish two distributions with certainty. In contrast, the more common metric of statistical distance only measures the probability of an adversary guessing correctly, even if the adversary may not know which of her guesses are correct. Then using Pinsker's inequality, we derive the Multisample Distinguisher Theorem, which bounds the multisample statistical distance as a function of the number of samples, the single-sample cover difference, and the single-sample KL-divergence.

The key insight that we obtain from the Multisample Distinguisher Theorem is as follows. For any padding distribution, the single-sample statistical distance can be decomposed into two disjoint components, of which one is the single-sample cover difference. These two components can then be translated to the q-sample setting by multiplying the cover difference by q and the remaining component by $\sqrt{q}$. That is, the cover difference component scales up more rapidly with the number of samples than the other component. In the particular case of uniform random padding, we observe that the single-sample statistical distance consists entirely of cover difference and therefore scales up linearly with the number of samples—which is the worst possible rate. Thus while uniform padding is a nearly optimal choice in the single-sample setting, it quickly loses its edge already when a small number of samples is available to the adversary.

Furthermore, the Multisample Distinguisher Theorem comes in handy when bounding the multisample statistical distance between two distributions. For instance, while the statistical distance can often be evaluated numerically for a single sample, evaluating it over multiple samples becomes quickly intractable as the number of samples increases. At the same time, it yields a fairly tight bound.

Gaussian Padding. The above analysis indicates that a suitable padding distribution must strike the right balance between minimising its single-sample statistical distance and minimising the single-sample cover difference. A small statistical distance ensures a good level of security already in the single sample setting, whereas a small cover difference ensures that the q-sample statistical distance scales up proportionally to $\sqrt{q}$ rather than q.

We identify two such padding distributions: the discrete Laplace and Gaussian distributions. We then study the security of these distributions in more detail through a combination of theoretical and empirical analysis. The Gaussian distribution emerges as the preferred choice. We evaluate its security for practical parameters and show that it offers significantly better performance than the uniform distribution in a setting where multiple samples are available to the adversary. In the appendix, we further compare the efficacy of these three padding schemes in thwarting the CRIME/BREACH attack. As we did above, one can compare the success probabilities and the number of encryptions for the distinct padding schemes in the context of the CRIME attack. Alternatively, we can consider the amount of overhead required by each scheme to attain a certain level of protection. Our results indicate that uniform padding would require a per-message overhead of roughly $20,000$ bytes to attain roughly the same level of protection that Gaussian padding

provides with just 200 bytes of overhead. As such, while Gaussian padding does not hide message lengths in an absolute sense, its performance benefits over uniform padding are substantial.

1.2 Length Hiding vs Fingerprinting

An adversary may benefit from learning the sizes of messages in various ways, depending on the setting at hand. Accordingly, we make a distinction between length-hiding security, the scope of this work, and fingerprinting security as covered in [18, 19], for instance. In fingerprinting, the ciphertext length is only one of the features of the encrypted traffic that enables the adversary to determine its origin. An adversary can additionally measure the total volume and the time intervals between consecutive ciphertexts in order to fingerprint the traffic. Indeed [18] indicates that length padding by itself is not very effective against website fingerprinting, exactly for this reason.

Despite this, we believe that length-hiding security still merits attention for at least two reasons. Firstly, there are other types of length attacks besides fingerprinting, such as CRIME and BREACH [20, 42], which are significantly different from fingerprinting. In particular, the total volume and the timing information are of little use here. On the other hand, the adversary requires the ability to choose messages which is typically not needed for website fingerprinting. Secondly, any length-hiding mechanism that we identify or improve upon could potentially serve as a stepping stone towards an (improved) aggregate countermeasure against fingerprinting. Thus, we emphasise that our treatment targets a scenario akin to CRIME or BREACH and that, by itself, it should not be considered in the context of fingerprinting.

2 PRELIMINARIES

2.1 Notation

Unless otherwise stated, an algorithm may be randomised. An adversary is an algorithm. For any adversary $\mathcal{A}$ and algorithms $\mathcal{X}, \mathcal{Y}, \ldots$ we use $\mathcal{A}^{\mathcal{X}(\cdot),\mathcal{Y}(\cdot),\cdots} \Rightarrow z$ to denote the process of running $\mathcal{A}$ with fresh coins and oracle access to algorithms $\mathcal{X}, \mathcal{Y}, \ldots$ and returning an output z. By convention the running time of an adversary refers to the sum of its actual running time and the size of its description. We generically refer to the resources of an adversary as any subset of the following quantities: its running time, the number of queries that it makes to its oracles, and the total length (in bytes) of its oracle queries.

If $\mathcal{S}$ is a set then $|\mathcal{S}|$ denotes its size, and $y \twoheadleftarrow \mathcal{S}$ denotes the process of selecting an element from $\mathcal{S}$ uniformly at random and assigning it to y. We use $\{0,1\}^n$ to denote the set of all binary strings of length n, and $\{0,1\}^*$ denotes the set of all binary strings of finite length. The empty string is represented by ε. For any two strings u and v, $|u|$ denotes the length of u in bytes and $u \| v$ denotes their concatenation.

We say that a probability distribution is discrete if its sample space is finite or countably infinite. For discrete probability distributions M and N over a sample space Ω, we use $\langle M \rangle$ to denote the support of M, and $M \times N$ to denote the distribution of pairs of values sampled independently from M and N. Similarly, we write M^q to denote the distribution of q independent and identically-distributed (i.i.d.) samples. Thus for any $\vec{x} = [x_1, \ldots, x_q] \in \Omega^q$, we have that $M^q(\vec{x}) = M(x_1) \cdots M(x_q)$.

2.2 Difference Measures.

Let M and N be two discrete distributions over a sample space Ω. The *statistical distance* between M and N is defined as:

$$\mathrm{SD}(M,N) = \sum_{x \in \Omega} \frac{1}{2} |M(x) - N(x)| = \sum_{x \in \Omega} \max\left(M(x) - N(x), 0\right) .$$

If N has full support, i.e. $\langle N \rangle = \Omega$, the *Kullback-Leibler (KL)* divergence between M and N is defined as:

$$\mathrm{D}(M \| N) = \sum_{x \in \Omega} M(x) \ln \left[\frac{M(x)}{N(x)} \right] .$$

While statistical distance satisfies the triangle inequality, KL divergence does not. Nevertheless KL divergence is *subadditive*, i.e.,

$$\mathrm{D}(M \times \widehat{M} \| N \times \widehat{N}) \leq \mathrm{D}(M \| N) + \mathrm{D}(\widehat{M} \| \widehat{N}) .$$

The subadditivity of KL divergence is a simple consequence of its chain rule [39]. In addition statistical distance and KL divergence are related via the well-known inequality attributed to Pinsker:

Lemma 2.1 (Pinsker's inequality [39]).

$$\mathrm{SD}(M,N)^2 \leq \frac{1}{2} \cdot \mathrm{D}(M \| N) .$$

Finally, while we will mostly be concerned with discrete distributions, we will also make use of the KL divergence for continuous distributions. For real-valued continuous distributions $\bar{M}$ and $\bar{N}$, where $\bar{N}$ has full support, the KL divergence is defined as:

$$\mathrm{D}(\bar{M} \| \bar{N}) = \int_{-\infty}^{\infty} \bar{M}(t) \ln \left[\frac{\bar{M}(t)}{\bar{N}(t)} \right] dt .$$

2.3 Symmetric Encryption

For added generality, we adopt the syntax of *Subtle AE* used in [6]. This allows us to additionally model decryption leakage originating from distinguishable decryption failures [8] or release of unverified plaintext [4].

A *symmetric encryption scheme* $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is a triple of algorithms such that:

- The randomised key generation algorithm $\mathcal{K}$ takes no input and returns a secret key K of fixed size. We will slightly abuse notation and use $\mathcal{K}$ to also identify the key space associated to the key generation algorithm.
- The encryption algorithm $\mathcal{E} : \mathcal{K} \times \{0,1\}^* \to \{0,1\}^*$, may be randomised, stateful or both. It takes as input the secret key $K \in \mathcal{K}$, a plaintext message $m \in \{0,1\}^*$, and returns a ciphertext in $\{0,1\}^*$. For stateful versions it may update its internal state when executed.
- The decryption algorithm $\mathcal{D} : \mathcal{K} \times \{0,1\}^* \to (\{\top, \bot\} \times \{0,1\}^*)$ is deterministic and may be stateful. It takes the secret key K and a ciphertext $c \in \{0,1\}^*$, to return a tuple (v, m) such that $v \in \{\top, \bot\}$ indicates the validity of the corresponding ciphertext, and m is a binary string representing a message or some leakage. It may update its state upon execution.

Note that decryption may either return $(\top, m)$, indicating that the ciphertext was valid and decrypts to the message $m \in \{0,1\}^*$, or $(\bot, m)$, indicating that the ciphertext was invalid where $m \in \{0,1\}^*$ may represent an error message, some internal value, or some other form of leakage. The leakage-free setting is modelled by returning $(\bot, \varepsilon)$ in response to an invalid ciphertext.

We further require that a symmetric encryption scheme satisfy the standard correctness condition stated below. As shorthand, we write $c_1, \ldots, c_n \leftarrow \mathcal{E}_K(m_1, \ldots, m_n)$ to denote the (in-order) sequence of encryption operations $c_1 \leftarrow \mathcal{E}_K(m_1), c_2 \leftarrow \mathcal{E}_K(m_2), \ldots, c_n \leftarrow \mathcal{E}_K(m_n)$. Similarly, $(v_1, m_1'), \ldots, (v_n, m_n') \leftarrow \mathcal{D}_K(c_1, \ldots, c_n)$ denotes the analogous sequence of decryption operations.

Definition 2.2 (Correctness). For all keys $K \in \mathcal{K}$, all $n \in \mathbb{N}$, and all message sequences $m_1, \ldots, m_n$, it must hold that if $c_1, \ldots, c_n \leftarrow \mathcal{E}_K(m_1, \ldots, m_n)$ and $(v_1, m_1'), \ldots, (v_n, m_n') \leftarrow \mathcal{D}_K(c_1, \ldots, c_n)$, then $v_i = \top$ and $m_i' = m_i$ for all $1 \le i \le n$.

We only require decryption to recover the honestly generated messages when ciphertexts are decrypted in the same order as they were produced. This slightly weaker correctness requirement allows us to cater for schemes with a stateful decryption algorithm.

2.4 Channel Simulatability

Various security notions exist for symmetric encryption, but we will make use of *channel simulatability* from [14]. In rough terms, it requires that for a secure scheme, some algorithm $\mathcal{S}$ can simultaneously simulate access to both encryption and decryption. Then, intuitively, access to the communication channel does not aid an adversary in any way since she can simulate it by herself. For technical reasons [14], the simulator algorithm $\mathcal{S}$ is augmented with a separate *wrapper* algorithm W that overwrites some of the simulator's decryption outputs.

Note that the standard notion of authenticated encryption (with pseudorandom ciphertexts) already implies channel simulatability. In fact, channel simulatability is merely a weaker generalisation thereof which does not mandate integrity, supports multiple errors and other forms of decryption leakage, and does not require ciphertexts to be pseudorandom[1]. We will use channel simulatability as a stepping stone to reach our end goal, rather than make it the end goal itself. Consequently, our use of channel simulatability as a preqrequisite only adds generality to our treatment. Below is the formal definition.

Definition 2.3 (Channel Simulatability). Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme. For any adversary $\mathcal{A}$ and a channel simulator $\mathcal{S}$ we define the corresponding CS advantage as:

$$\mathsf{Adv}^{\mathsf{cs}}_{\mathcal{SE}}(\mathcal{A}, \mathcal{S}) = \Pr\left[\, \mathcal{A}^{\mathcal{E}_K(\cdot), \mathcal{D}_K(\cdot)} \Rightarrow 1 \,\right] - \Pr\left[\, \mathcal{A}^{\mathcal{S}(\mathsf{e}, |\cdot|), \mathsf{W}[\mathcal{S}](\mathsf{d}, \cdot)} \Rightarrow 1 \,\right],$$

where probabilities are taken over $K \twoheadleftarrow \mathcal{K}$ and the algorithms' coin tosses. A scheme $\mathcal{SE}$ is said to be $(\epsilon, \mathcal{R}_{\mathcal{S}}, \mathcal{R}_{\mathcal{A}})$-CS secure, if there exists a randomised and possibly stateful simulator $\mathcal{S}$ such that every query of the form $\mathcal{S}(\mathsf{e}, \cdot)$ or $\mathcal{S}(\mathsf{d}, \cdot)$ requires at most $\mathcal{R}_{\mathcal{S}}$ resources, and for any adversary $\mathcal{A}$, requiring at most $\mathcal{R}_{\mathcal{A}}$ resources, its respective advantage $\mathsf{Adv}^{\mathsf{cs}}_{\mathcal{SE}}(\mathcal{A}, \mathcal{S})$ is bounded by ϵ.

[1] A stronger variant of channel simulatability with integrity is possible [14], but we will not need this here.

$\mathsf{W}[\mathcal{S}](\mathsf{d}, c')$	$\overline{\mathsf{W}}[\mathcal{S}](\mathsf{d}, c')$
$(v, m') \leftarrow \mathcal{S}(\mathsf{d}, c')$	$(v, m') \leftarrow \mathcal{S}(\mathsf{d}, c')$
if $\exists\, m$ s.t. $(m, c') \in \mathsf{T}$	**if** $\exists\, m$ s.t. $(m, c') \in \mathsf{T}$
$(v, m') \leftarrow (\top, m)$	$(v, m') \leftarrow (\lightning, \lightning)$
return (v, m')	**return** (v, m')

Figure 1: Left: The wrapper W used to define channel simulatability. Right: A similar wrapper $\overline{\mathsf{W}}$ which suppresses output instead of replacing it with transcript values.

Let us now unpack the above definition. Throughout the adversary's interaction with its encryption oracle (real or simulated), a transcript T is maintained of its queries and responses. This is simply a list of message-ciphertext pairs (m, c), where each entry corresponds to an encryption query. As for the simulator $\mathcal{S}$, note that it is a *single* algorithm with separate interfaces for encryption and decryption. That is, $\mathcal{S}(\mathsf{e}, |\cdot|)$ and $\mathcal{S}(\mathsf{d}, \cdot)$ share the same state and randomness. Note that the simulator is only fed the size of the message in an encryption query. Furthermore, access to simulated decryption queries is mediated through a *wrapper* algorithm W that forwards queries between the adversary and the simulator, and possibly overwrites the output of the simulator. Namely the wrapper will detect whether a ciphertext corresponds to a prior encryption query and replace the output of $\mathcal{S}$ with the message in the transcript. The simulator is unaware of the wrapper's actions. The wrapper is needed because if the simulator were to be given direct access to the transcript the resulting security notion would no longer guarantee confidentiality [14]. Due to the presence of W, there is no need to restrict the adversary's decryption queries as in other security definitions. A pseudocode description of W is displayed in Figure 1. In some proofs we will make use of a second wrapper $\overline{\mathsf{W}}$, also shown in Figure 1, which operates similarly but suppresses output instead of overwriting it.

3 LENGTH HIDING SECURITY

We now introduce our formal security definition for length hiding and show that a symmetric encryption scheme that is channel simulatable can be safely composed together with an appropriate padding scheme to meet our notion. This simple composition theorem allows us to reduce length hiding security to a purely information-theoretic problem, namely that of distinguishing between two distributions over the integers which we can sample multiple times.

3.1 Prior Security Definitions

Our Length Hiding security definition builds on two prior security definitions that were proposed independently in PRS11 and TV11. An overview of these works is provided in Appendix A where we discuss their limitations in more detail. Comparing the definitions from PRS11 and TV11 we see good and bad aspects in both. On the

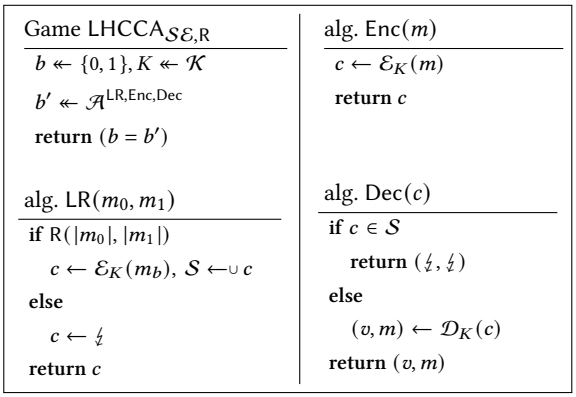

Figure 2: The Length Hiding Game.

one hand the PRS11 definition captures the fact that length information might leak at the cryptographic level, say during decryption. In contrast, the TV11 definition adopts a more simplistic formulation that does not take this into account which excludes access to the decryption algorithm and allows only a single encryption query. However, unlike PRS11, their security definition takes the padding strategy into account. Our length-hiding security definition takes the best of both worlds and merges these into a stronger definition. In addition by parametrising security via a relation that is independent of the scheme, we avoid a circularity issue present in the PRS11 definition, where the degree of length-hiding provided by the notion is dependent on the scheme.

3.2 A Unified Security Definiton

In view of the limitations in the security models put forth by PRS11 and TV11, we propose a unified security definition. Like PRS11 our definition allows the adversary to make multiple encryption and decryption queries. However, as in TV11, the ciphertext length is determined by the scheme rather than the adversary, which allows us to evaluate the length hiding mechanism, i.e., the padding strategy. Unlike PRS11, however, our notion does not guarantee integrity as we view length hiding being akin to confidentiality but separate from integrity as a goal. Below is the formal definition.

Definition 3.1 (Length Hiding Security). Let R be some fixed relation over pairs of non-negative integers. Then for any symmetric encryption scheme $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ and any adversary $\mathcal{A}$ we define the corresponding LH-CCA advantage as:

$$\mathsf{Adv}^{\mathsf{lh\text{-}cca}}_{\mathcal{SE},\mathsf{R}}(\mathcal{A}) = 2\Pr[\mathsf{LHCCA}^{\mathcal{A}}_{\mathcal{SE},\mathsf{R}} \Rightarrow \mathsf{true}] - 1\,.$$

For generality our security notion is parametrised by a relation R over message-length pairs. This replaces the usual condition that both messages be of the same length and allows us to tune our security definition to different 'flavours' of length hiding – see the next paragraph on deterministic padding for examples. However, our focus will be on the relation R_Δ where $\mathsf{R}_\Delta(|m_0|, |m_1|) = \mathsf{true}$ if and only if $0 \le |m_1| - |m_0| \le \Delta_{max}$, for some positive integer Δ_{max}. The adversary is given access to a challenge oracle as well as a plain encryption oracle. This allows us to separate challenge queries from encryption queries in the security bound. Note that both oracles make use of the same instance of the encryption algorithm, i.e., the two oracles share the state of $\mathcal{E}_K$. Note also that the lengths of the messages can vary from query to query as long as the relation R_Δ is satisfied.

3.3 Security Model and Deterministic Padding

A number of deterministic padding schemes have been suggested in the literature – see [18] for some examples. One such example is the *padding by rounding* technique, typically employed in CBC encryption, in which the padding length is selected so as to extended the message to fill the nearest block. The padding technique employed in Tor can also be viewed as an instance of padding by rounding. Any such deterministic padding scheme that permits more than one ciphertext length offers no protection in our security model. This is because in our security model the adversary has control over the message lengths. Then for any such padding scheme the adversary can always, even when $\Delta = 1$, choose a pair of messages whose lengths lie on either side of the threshold value thereby resulting in two ciphertexts of distinct lengths.

As we already pointed out in Section 1.2 allowing adversarial control over the message lengths is necessary to capture length-based attacks such as CRIME and BREACH [20, 42]. Accordingly, in this paper we focus on randomised length padding. This does not imply, however, that deterministic padding may not be beneficial in other settings. In fact, recent work by Gellert, Jager, Lyu, and Neuschulten [19] argues in favour of deterministic padding in the context of fingerprinting. Besides the adversarial control over message lengths, they point to the fact that our security model focuses on distinguishing between the encryptions of two messages whereas theirs extends to a setting with more messages. Essentially, in our security model the adversary is challenged to guess a single bit of information whereas in theirs there may be more information for the adversary to learn in order to win. As such, while these differences effectively weaken the security definition they seem more appropriate for the fingerprinting scenario. On the other hand, if the goal is to protect against attacks like CRIME and BREACH our security definition is a better choice.

Finally, we note that it is also possible to tune our security notion to be more amenable to deterministic padding by considering a different relation other than R_Δ. Namely, replacing it with a relation that holds true for any pair of messages whose lengths map to the same value when rounded to the nearest block. Indeed that security definition would be analogous to the LHAE definition of PRS11 [36] but would avoid its circularity issue (see Appendix A.1).

3.4 Reduction to Information-Theoretic Setting

The attacks presented in [15, 36] against TLS and IPsec show that combining length padding and encryption is not as straightforward as one might think. In particular the cryptographic component might leak information about the padding length even if the message contents are not leaked. Thus no matter how good the statistical guarantees of the length padding may be, it is all in vain if the resulting composition does not resist such attacks. We present a simple composition theorem showing that appending a message with padding before encryption is safe if the encryption scheme is

Game $\mathsf{DHIDE}_{\mathcal{P},\Delta_{max}}$	alg. Hide(Δ)
$b \leftarrow \{0,1\}$	**if** $\Delta \le \Delta_{max}$
$b' \leftarrow \mathcal{A}^{\mathsf{Hide}}$	$z \leftarrow \mathcal{P}$
return $(b = b')$	**if** $b = 1$ **then** $z \leftarrow z + \Delta$
	else
	$z \leftarrow \lightning$
	return z

Figure 3: The Difference Hiding Game.

channel simulatable. Indeed, looking back at the attacks in [15, 36] we observe that the schemes employed in TLS 1.2 and IPsec in those situations were not channel simulatable. Nevertheless, most Authenticated Encryption schemes in use today, such as GCM [30] and ChaCha-Poly1305 [34], do meet this notion. Another way to view our composition theorem is that channel simulatability allows us to reduce LH-CCA security to the statistical properties of the probability distribution by which the padding length is determined. More specifically, for a padding distribution $\mathcal{P}$, the statistical properties which we require are captured by the Difference Hiding notion described below.

Definition 3.2 (Difference Hiding). Let $\mathcal{P}$ be a padding distribution. Then for any adversary $\mathcal{A}$ we define the corresponding D-HIDE advantage as:

$$\mathsf{Adv}^{\mathsf{d\text{-}hide}}_{\mathcal{P},\Delta_{max}}(\mathcal{A}) = 2\Pr[\mathsf{DHIDE}^{\mathcal{A}}_{\mathcal{P},\Delta_{max}} \Rightarrow \mathsf{true}] - 1\,.$$

In the above game the adversary interacts with a Hide oracle. This oracle takes as input a non-negative integer Δ, smaller or equal to some maximum value Δ_{max}, it then samples another non-negative integer z from $\mathcal{P}$, and returns either z or $z + \Delta$ with equal probability. The goal of the adversary is to distinguish between these two cases.

Note that the D-HIDE security of the padding scheme depends solely on the sampling algorithm $\mathcal{P}$ which determines the length of the padding. However, a padding scheme requires two additional algorithms, pad and unpad, satisfying a simple correctness requirement. The former is a possibly randomised algorithm taking as input an integer z and returning a padding string of size z. Then for any message, whenever such a padding string is appended to it and fed into unpad the original message will be returned as the output. We are now ready to state our composition theorem.

alg. $\overline{\mathcal{E}}(m)$	alg. $\overline{\mathcal{D}}(c)$
$z \leftarrow \mathcal{P}$	$w \leftarrow \mathcal{D}_K(c)$
$c \leftarrow \mathcal{E}_K(m \parallel \mathsf{pad}(z))$	$m \leftarrow \mathsf{unpad}(w)$
return c	**return** m

Figure 4: Padded Encryption Scheme.

Theorem 3.3 (Padding Composition). *For any symmetric encryption scheme $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ and any padding scheme $(\mathcal{P}, \mathsf{pad}, \mathsf{unpad})$, let $\overline{\mathcal{SE}} = (\mathcal{K}, \overline{\mathcal{E}}, \overline{\mathcal{D}})$ be the composed encryption scheme described in Figure 4. Then for any positive integer Δ_{max} associated to the relation R_Δ, and any* LH-CCA *adversary $\mathcal{A}$, there exists a corresponding* D-HIDE *adversary $\mathcal{A}_{dh}$, and a corresponding* CS *adversary $\mathcal{A}_{cs}$ such that for all simulators $\mathcal{S}$*

$$\mathsf{Adv}^{\mathsf{lh\text{-}cca}}_{\overline{\mathcal{SE}},\mathsf{R}_\Delta}(\mathcal{A}) \le \mathsf{Adv}^{\mathsf{cs}}_{\mathcal{SE}}(\mathcal{A}_{cs}, \mathcal{S}) + \mathsf{Adv}^{\mathsf{d\text{-}hide}}_{\mathcal{P},\Delta_{max}}(\mathcal{A}_{dh})$$

(Proof Sketch). The proof proceeds as follows. We make a single game hop from the LHCCA game to a new game G, and then reduce this game to the DHIDE game. The game G is simply the LHCCA game where calls to $\mathcal{E}_K(\cdot)$ and $\mathcal{D}_K(\cdot)$ are replaced with calls to $\mathcal{S}(\mathsf{e}, |\cdot|)$ and $\overline{\mathsf{W}}[\mathcal{S}](\cdot)$, respectively. Note that we make use of $\overline{\mathsf{W}}$ instead of W since the LH-CCA game suppresses output. Then games LHCCA and G are indistinguishable up to the channel simulatability bound. Now G can be simulated via the DHIDE game as follows. For every left-or-right query (m_0, m_1) the difference Δ is set equal to $|m_1| - |m_0|$ and queried to the Hide oracle to obtain a value z. The ciphertext is then evaluated by feeding the value $(z + |m_0|)$ to $\mathcal{S}(\mathsf{e}, |\cdot|)$. This provides a perfect simulation of game G and thus the advantage in winning the DHIDE game is at least that of winning game G. □

4 KNOWING WHEN YOU'RE RIGHT

In most cryptographic settings security often translates to an adversary's success probability being negligible. In our setting such a requirement is out of reach as it would drastically impact the overall efficiency of our system and we have to content ourselves with the adversary's probability of success being only moderately small. In an information theoretic setting the security measure is usually expressed in terms of the statistical distance between two distributions. However the fact that we now have to deal with non-negligible success probabilities prompts us to reconsider whether statistical distance is comprehensive enough as a measure. In particular, as a figure, statistical distance reflects only the adversary's probability of guessing correctly and ignores whether the adversary is able to determine if it has guessed correctly or not. When the statistical distance is negligible there is no point in discerning between these two cases. In contrast, when the probability of success is non-negligible, limiting the adversary's ability to determine whether it guessed correctly or not can in turn limit its ability to act on that information and potentially curb its overall efficacy.

In the rest of this section, we introduce a new simple measure, which we call *cover difference*, that quantifies the probability of an adversary making a correct guess with certainty. We propose the *combination* of statistical distance and cover difference as a more comprehensive measure of security in this kind of settings. We will show that, quite naturally, the cover difference is bounded above by the statistical distance. Accordingly, an interesting measure is the *ratio* of the cover difference to the statistical distance, representing the proportion of times the adversary can recognise a distribution with certainty out of the number of times it makes a correct guess. Then, given two countermeasures offering similar bounds in terms of statistical distance but significantly different values for the cover difference, the one with the lower cover difference should emerge as the preferable choice. In addition, as we will show later in this section, the ratio of cover difference to statistical distance conveys

information regarding the rate at which the statistical distance scales up with the number of samples available to the adversary. Namely, the bigger the ratio, the more rapidly an adversary's guessing ability will increase with the number of available samples. Thus, in the example above, the countermeasure with the lower cover difference will perform better in terms of statistical distance over multiple samples, even if they offer the same statistical distance for a single sample. We emphasise that statistical distance remains our primary security measure, and cover difference will serve only as an analytical tool. Before delving into the technical details, we provide some insight into why an attacker's ability to ascertain the veracity of its hypotheses is relevant in practice.

Let us start by pointing out that certain attacks are qualified as such, not because they have a success probability higher than what is considered inevitable, but precisely because they allow the adversary to be certain about her success. Consider, for example, the simplest variant of the SSH attack in [2] whereby an adversary is able to verifiably recover 14 bits of plaintext with a probability of 2^{-14}. Now, irrespective of how good the encryption is, an adversary can always guess 14 bits of plaintext with a probability of 2^{-14}. That is, this attack does no better, in terms of success probability, than an attacker that merely outputs 14 random bits. However, by interacting with the decryption algorithm and observing its error messages, the attacker is able to obtain confirmation of her success. As a second example, let us examine, from an attacker's perspective, how this aspect plays out in attacks which proceed in an iterative fashion. Instances of such attacks are padding oracle attacks such as those appearing in [1, 3, 11, 15, 35, 50] and Bleichenbacher's attack on PKCS#1v1.5 [7, 31]. These attacks progress in stages, where in each stage the adversary makes some queries to an oracle. However, what gets queried in each stage depends on the responses of the oracle in the prior stage. These attacks typically require precise information from the oracle to succeed, and thus they can be adversely affected if the oracle in question is 'noisy'. Often the oracle's output is a single bit determined by the magnitude of a timing difference or the difference in the length of an encrypted error message. Any single error at any stage will cause the attack to return the wrong output. Thus, even with relatively low noise, the chance of the attack returning a wrong output can be quite high as the probability of error aggregates over multiple stages. In these cases the attacker's ability to obtain certainty with high probability as opposed to guessing correctly with high probability becomes very valuable. A similar situation arises in volume attacks against databases [21, 25] where noisy observations can be very detrimental.

4.1 The Cover Difference

We introduce the *cover difference* as a measure of an adversary's ability to distinguish between two distributions *with certainty*. Informally it can be described as the average probability that when one of two distributions is sampled, the sample value falls outside the support of the other distribution. Thus since the sample value can only have originated from one of the two distributions, in such a case the adversary can distinguish with zero chance of error. The formal definition is given below.

Definition 4.1 (Cover Difference). For two discrete distributions M and N over a sample space Ω, their cover difference is defined as:

$$\mathrm{CD}(M,N) = \sum_{x\in\langle M\rangle\setminus\langle N\rangle}\frac{M(x)}{2} + \sum_{x\in\langle N\rangle\setminus\langle M\rangle}\frac{N(x)}{2} = \sum_{x\in\langle M\rangle\triangle\langle N\rangle}\frac{1}{2}\left[M(x)+N(x)\right].$$

Note that cover difference is not a metric since $\mathrm{CD}(M,N)=0$ does not imply that $M=N$, and it does not satisfy the triangle inequality[2]. Nevertheless it satisfies the following two useful properties.

Lemma 4.2. *Let $M, \widetilde{M}, N, \widetilde{N}$ be discrete probability distributions over a sample space Ω. It then holds that:*

(a) The cover difference is bounded above by the statistical distance, i.e.

$$\mathrm{CD}(M,N) \le \mathrm{SD}(M,N).$$

(b) The cover difference is subadditive, i.e.

$$\mathrm{CD}(M\times\widehat{M}, N\times\widehat{N}) \le \mathrm{CD}(M,N) + \mathrm{CD}(\widehat{M},\widehat{N}).$$

Proof. To prove the first part of the Lemma we start from the formula for calculating the cover difference

$$\mathrm{CD}(M,N) = \sum_{x\in\langle M\rangle\setminus\langle N\rangle}\frac{M(x)}{2} + \sum_{x\in\langle N\rangle\setminus\langle M\rangle}\frac{N(x)}{2}.$$

Since $N(x)=0$ for $x\in\langle M\rangle\setminus\langle N\rangle$ and $M(x)=0$ for $x\in\langle N\rangle\setminus\langle M\rangle$ we can replace the summands to obtain

$$= \sum_{x\in\langle M\rangle\setminus\langle N\rangle}\frac{|M(x)-N(x)|}{2} + \sum_{x\in\langle N\rangle\setminus\langle M\rangle}\frac{|M(x)-N(x)|}{2}.$$

Now, the above expression is similar to that for statistical distance except that the summation is over $\langle M\rangle\triangle\langle N\rangle$ instead of Ω. Then, since $\langle M\rangle\triangle\langle N\rangle \subseteq \langle M\rangle\cup\langle N\rangle \subseteq \Omega$ and all summands are positive, the claim follows, i.e.,

$$\le \sum_{x\in\Omega}\frac{1}{2}|M(x)-N(x)| = \mathrm{SD}(M,N).$$

We now move on to the second part of the Lemma. Evaluating the left hand side using the formula yields

$$\mathrm{CD}(M\times\widehat{M}, N\times\widehat{N}) = \sum_{(x,y)\in\langle M\times\widehat{M}\rangle\setminus\langle N\times\widehat{N}\rangle}\frac{M(x)\widehat{M}(y)}{2} + \sum_{(x,y)\in\langle N\times\widehat{N}\rangle\setminus\langle M\times\widehat{M}\rangle}\frac{N(x)\widehat{N}(y)}{2}. \quad (1)$$

We now make use of the fact that $\langle M\times\widehat{M}\rangle\setminus\langle N\times\widehat{N}\rangle$ can be expressed as the union of $\langle M\rangle\setminus\langle N\rangle\times\langle\widehat{M}\rangle$ and $\langle M\rangle\times\langle\widehat{M}\rangle\setminus\langle\widehat{N}\rangle$. Then the first term on the right hand side corresponds to the probability of (x,y) being contained in this union. Applying the union bound to the first term yields

$$\sum_{(x,y)\in\langle M\times\widehat{M}\rangle\setminus\langle N\times\widehat{N}\rangle}\frac{M(x)\widehat{M}(y)}{2} \le \sum_{(x,y)\in\langle M\rangle\setminus\langle N\rangle\times\langle\widehat{M}\rangle}\frac{M(x)\widehat{M}(y)}{2} + \sum_{(x,y)\in\langle M\rangle\times\langle\widehat{M}\rangle\setminus\langle\widehat{N}\rangle}\frac{M(x)\widehat{M}(y)}{2}.$$

[2]Which is why we choose to call it a difference rather than a distance. Besides, the term also hints to its relation to the set difference between supports.

Expanding each summation on the right into two summations over single variables and then simplifying, we obtain

$$= \sum_{x \in \langle M \rangle \setminus \langle N \rangle} \frac{M(x)}{2} + \sum_{y \in \langle \widehat{M} \rangle \setminus \langle \widehat{N} \rangle} \frac{\widehat{M}(y)}{2} . \tag{2}$$

Applying a similar argument to the second term on the right hand side of (1) yields

$$\sum_{(x,y) \in \langle N \times \widehat{N} \rangle \setminus \langle M \times \widehat{M} \rangle} \frac{N(x)\widehat{N}(y)}{2} \leq \sum_{x \in \langle N \rangle \setminus \langle M \rangle} \frac{N(x)}{2} + \sum_{y \in \langle \widehat{N} \rangle \setminus \langle \widehat{M} \rangle} \frac{\widehat{N}(y)}{2} . \tag{3}$$

By combining (1), (2), and (3), and grouping terms together we obtain the claimed result. □

4.2 Distinguishing With Multiple Samples

An adversary's ability to distinguish between two distributions depends on the number of independent samples available to the adversary. In general we expect that the more samples become available the better its ability to distinguish. Less obvious, however, is how its distinguishing ability scales with the number of samples. Below we derive an upper bound on the statistical distance for multiple samples in terms of the KL-divergence and cover difference for a single sample and the number of samples. This bound is stated formally in Theorem 4.3. Recall that the statistical distance represents the best possible distinguishing advantage of any (unbounded) adversary. Consequently, Theorem 4.3 yields an upper bound on the distinguishing advantage of any multisample distinguisher. Intuitively, the right hand side of inequality (5) in Theorem 4.3 decomposes the single-sample difference between two distributions into two mutually exclusive components: the KL-divergence and the cover difference. Specifically, the KL-divergence component is evaluated over the points contributing to the statistical distance between the two distributions that do not contribute to the cover difference. Then, the notable feature that emerges from Theorem 4.3 is that the bound on the cover difference component is linear in the number of i.i.d. samples q, whereas the bound on the remaining component is proportional to $\sqrt{q}$. Furthermore, in Section 4.3 we will show that the linear bound on the cover difference is tight for the range of values that we are interested in. Hence, we see that the statistical distance grows (at most) proportionately to $\sqrt{q}$ except for the cover difference component. Accordingly, in order to limit the adversary's benefit from the multiple samples that it may gather, we must keep the cover difference component to a minimum.

Theorem 4.3 (Multisample Distinguisher). *Let M and N be two discrete probability distributions over a sample space Ω. Consider then an experiment where an adversary attempts to distinguishing q i.i.d. samples drawn from one of these two distributions. It then holds that:*

$$\mathrm{CD}(M^q, N^q) \leq q \cdot \mathrm{CD}(M, N) \tag{4}$$

$$\mathrm{SD}(M^q, N^q) \leq 2q \cdot \mathrm{CD}(M, N) + \sqrt{\frac{q}{2} \cdot \mathrm{D}(\widetilde{M} \,\|\, \widetilde{N})}\,, \tag{5}$$

where

$$\widetilde{M}(x) = \begin{cases} 0 & : \quad x \in \langle M \rangle \setminus \langle N \rangle \\ \widetilde{A}_M \cdot M(x) & : \quad x \in \langle M \rangle \cap \langle N \rangle \end{cases}$$

$$\widetilde{N}(x) = \begin{cases} 0 & : \quad x \in \langle N \rangle \setminus \langle M \rangle \\ \widetilde{A}_N \cdot N(x) & : \quad x \in \langle N \rangle \cap \langle M \rangle \end{cases}$$

and $\widetilde{A}_M$ and $\widetilde{A}_N$ are real-valued normalising factors.

Proof of Theorem 4.3. Let $\widetilde{M}$ and $\widetilde{N}$ be as defined in the Theorem statement. Inequality (4) follows directly from repeated application of the subadditive property in Lemma 4.2, so we focus on deriving inequality (5). Repeated application of the triangle inequality yields

$$\mathrm{SD}(M^q, N^q) \leq \mathrm{SD}(M^q, \widetilde{M}^q) + \mathrm{SD}(N^q, \widetilde{N}^q) + \mathrm{SD}(\widetilde{M}^q, \widetilde{N}^q) . \tag{6}$$

Expanding the first term on the right-hand side we note that

$$\begin{aligned} \mathrm{SD}(M^q, \widetilde{M}^q) &= \sum_{\vec{x} \in \Omega^q} \max\left(M^q(\vec{x}) - \widetilde{M}^q(\vec{x}), 0\right) \\ &= \sum_{\vec{x} \in \langle M^q \rangle \setminus \langle N^q \rangle} M^q(\vec{x}) , \end{aligned} \tag{7}$$

since by definition $\widetilde{M}^q(\vec{x}) = 0$ for $\vec{x} \in \langle M^q \rangle \setminus \langle N^q \rangle$ and $\widetilde{M}^q(\vec{x}) \geq M^q(\vec{x})$ otherwise. Similarly, it follows that

$$\mathrm{SD}(N^q, \widetilde{N}^q) = \sum_{\vec{x} \in \langle N^q \rangle \setminus \langle M^q \rangle} N^q(\vec{x}) . \tag{8}$$

Then, by the definition of cover difference, adding up equations (7) and (8) yields

$$\mathrm{SD}(M^q, \widetilde{M}^q) + \mathrm{SD}(N^q, \widetilde{N}^q) = 2\mathrm{CD}(M^q, N^q) . \tag{9}$$

Now, note that $\langle \widetilde{N}^q \rangle = \langle \widetilde{M}^q \rangle = (\langle M \rangle \cap \langle N \rangle)^q$ and hence the value of $\mathrm{D}(\widetilde{M}^q \,\|\, \widetilde{N}^q)$ is finite. We can therefore apply Lemma 2.1 followed by the subadditivity of KL divergence to obtain

$$\begin{aligned} \mathrm{SD}(\widetilde{M}^q, \widetilde{N}^q) &\leq \sqrt{\frac{1}{2} \cdot \mathrm{D}(\widetilde{M}^q \,\|\, \widetilde{N}^q)} \\ &= \sqrt{\frac{q}{2} \cdot \mathrm{D}(\widetilde{M} \,\|\, \widetilde{N})} \end{aligned} \tag{10}$$

Combining (6), (9), and (10) yields the desired result. □

Remarks. Note that in the special case where $\langle M \rangle \subseteq \langle N \rangle$, we can express the bound in (5) directly in terms of $\mathrm{D}(M \,\|\, N)$. Here $\widetilde{M}$ reduces to M, since $\langle M \rangle \setminus \langle N \rangle = \emptyset$. Then we have that

$$\begin{aligned} \mathrm{D}(\widetilde{M} \,\|\, \widetilde{N}) &= \sum_{x \in \langle \widetilde{N} \rangle} M(x) \ln\left[\frac{M(x)}{\widetilde{N}(x)}\right] \\ &= \sum_{x \in \langle N \rangle} M(x) \left\{\ln\left[\frac{M(x)}{N(x)}\right] - \ln \widetilde{A}_N\right\} \\ &= \mathrm{D}(M \,\|\, N) - \ln[1 - 2\mathrm{CD}(M, N)] . \end{aligned}$$

Results similar in spirit to Theorem 4.3 have appeared in earlier works. In [40], Renner showed that under certain restricted conditions the statistical distance between two distributions grows proportionally to the square root of the number of samples. However the constant factor in the bound is proportional to the inverse of the minimum probability over all elements in the sample space. Since we will be concerned with distributions with arbitrarily small

probabilities this bound is too loose to be of any use. A very similar result to inequality (5) in Theorem 4.3 can be derived using the Chi-squared method [13] which combines Pinsker's inequality and the χ^2-divergence [33]. In our case, relying directly on the KL-divergence instead of the χ^2-divergence yields a better bound.

4.3 A Cover Difference Lower Bound

We now show that the elevated rate at which the cover difference component scales as a function of the number of samples, is in fact inherent and not merely an artefact of the bound derived in Theorem 4.3. Namely, there is a simple generic attack showing that the upper bound on the cover difference component is tight, especially when its value is small. The cover difference corresponds to the probability of sampling a value that automatically leaks from which distribution it was sampled, since that value is only contained in the support of one of the two distributions. Thus, when observing q i.i.d. samples, it suffices that just one sample is of this kind for the adversary to be able to determine with certainty the distribution from which the samples originated. Thus, the cover difference for the multisample distributions is simply the probability that at least one sample is of this kind. Therefore

$$\mathrm{CD}(M^q, N^q) = 1 - (1 - \mathrm{CD}(M, N))^q ,$$

where for sufficiently small values of $\mathrm{CD}(M, N)$ this expression is closely approximated by

$$\approx q \cdot \mathrm{CD}(M, N) .$$

5 OUTMATCHING UNIFORM PADDING

Informed by the analysis in Section 4 we are now better equipped to identify a suitable distribution for length padding. We start by examining the multisample security of the uniform distribution and expose its weaknesses.

5.1 The Problem with Uniform Padding

The Multisample Distinguisher Theorem suggests that we use a distribution such that the cover difference between this distribution and its shifted copy is as small as possible. This would ensure that the statistical distance would grow at most proportionally to $\sqrt{q}$ rather than linearly in q. This means that the adversary is not as effective in amplifying its advantage through multiple samples. In addition, besides reducing its chance of distinguishing correctly it also reduces its chance of distinguishing with certainty. Under this light we then see that uniform random padding performs rather poorly. Let $U(l)$ denote a discrete uniform distribution over the interval $[0, l]$, and $U_\Delta(l)$ denote the same distribution offset by Δ points to the right. In this case, we immediately observe that for any offset Δ, the statistical distance is comprised solely of the cover difference, i.e.,

$$\mathrm{SD}(U(l), U_\Delta(l)) = \mathrm{CD}(U(l), U_\Delta(l)) = \frac{\Delta}{l+1} .$$

Thus for a uniform distribution the cover difference takes on its maximum value possible. This is due to the fact that the values on which the two distributions differ are contained in exactly one of the supports, as shown in the example in Figure 5. This also

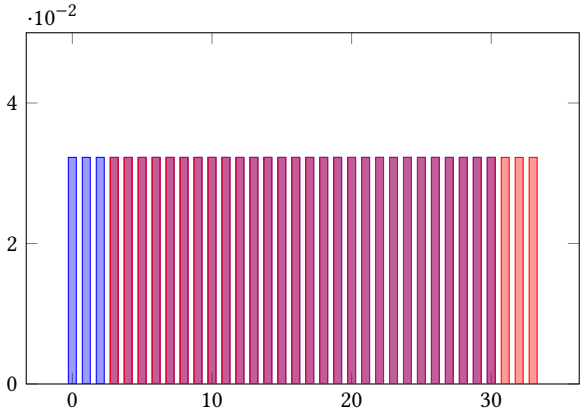

Figure 5: Illustrating the statistical distance between two uniform distributions. Blue: $U(30)$, Red: $U_\Delta(30)$, $\Delta = 3$.

means that the attack described in Section 4.3 is directly applicable to uniform padding. Reconciling this with the result by Tezcan and Vaudenay [47], we see that the uniform distribution achieves the best possible statistical distance for a single sample but it then increases at the worst possible rate as the number of samples is increased. In addition the distinguishing advantage corresponds exactly to the probability of the adversary distinguishing with certainty.

5.2 Searching for Alternatives

We seek to improve over the uniform distribution by lowering the cover difference, while maintaining a similar tradeoff between the average padding length and the (single sample) statistical distance for a given offset Δ. That way, we retain a comparable bound for statistical distance in the single-query setting, but we improve in terms of the cover difference as well as statistical distance in the multiple query setting. Intuitively, it is easy to see that the cover difference between a distribution and its shifted copy, is related to the tails of the distribution. Accordingly, distributions with their probability mass concentrated in the middle and small tails appear to be a favourable choice towards this goal. We examine two such distributions: the Laplace distribution and the Gaussian distribution.

Besides fitting the above profile the other reason for choosing these distributions is their suitability for Differential Privacy applications [17]. Indeed length hiding can be viewed as a specific instance of a Differential Privacy problem. Hiding a length difference is analogous to hiding the value corresponding to a single entry in a database from the sum of all entries in the database. On the other hand in our case we are not concerned with maintaining a certain level of accuracy in the statistic and we are also using a different security measure than differential privacy. Nevertheless this analogy makes the Laplace and Gaussian distributions natural candidates to consider for our setting.

5.3 Discrete Laplace Distribution

The discrete Laplace distribution $L(a, b)$, scaled by a factor a and centred at b, is given by:

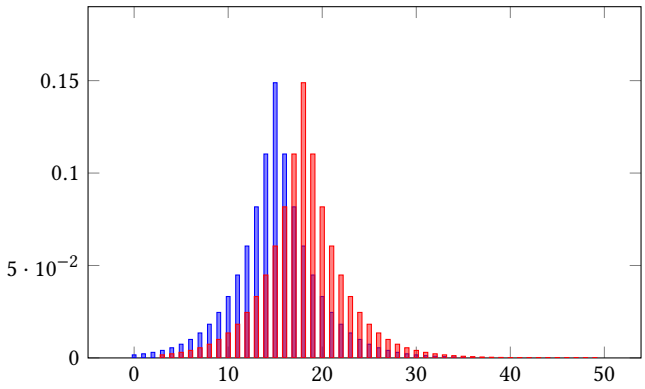

Figure 6: Illustrative example for the truncated discrete Laplace distribution. Blue: $\bar{L}(0.3, 15)$, Red: $\bar{L}_\Delta(0.3, 15)$, $\Delta = 3$.

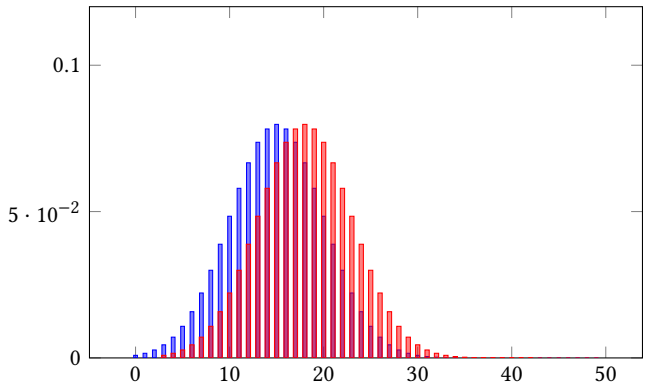

Figure 7: Illustrative example for the truncated discrete Gaussian distribution. Blue: $\bar{N}(15, 5)$, Red: $\bar{N}_\Delta(15, 5)$, $\Delta = 3$.

$$p_L(x) = \tanh\left(\frac{a}{2}\right) e^{-a|x-b|} = \left(\frac{e^a - 1}{e^a + 1}\right) e^{-a|x-b|}.$$

However this does not quite fit our purposes yet, so we will make use instead of a Laplace distribution that is truncated on the left side and rescaled accordingly. Alternatively one could also truncate both sides at the expense of doubling the cover difference. Since the cover difference will generally be quite small this should not affect our analysis significantly. Then for all $x \geq 0$, the (one-sided) truncated Laplace distribution $\bar{L}(a, b)$ is defined as:

$$p_{\bar{L}}(x) = A_L \tanh\left(\frac{a}{2}\right) e^{-a|x-b|} = \left[\frac{e^{ab}(e^a - 1)}{e^{ab}(e^a + 1) - 1}\right] e^{-a|x-b|},$$

where

$$A_L = \left(\frac{1}{1 - \frac{e^{-ab}}{e^a+1}}\right).$$

The truncated Laplace distribution is illustrated in Figure 6 with parameter $a = 0.3$ and $b = 15$. Now, as we vary the parameters (a, b) some opposing forces come into play. In order to draw a comparison with the uniform distribution we set these parameters as follows. When the area under the truncated part is small the average padding length approaches the value b. For a uniform distribution the average padding length is equal to:

$$\sum_{x=0}^{l} \frac{x}{l+1} = \frac{l(l+1)}{2(l+1)} = \frac{l}{2}.$$

Thus we start by setting $b = \frac{l}{2}$, so that both distributions result in roughly equal overhead, and then go on to adjust the value of a. Now, as a increases the area under the truncated part and the cover difference both decrease. To see this, note that the area under the truncated part is equal to $(1 - 1/A_L)$, and the cover difference is given by

$$\mathrm{CD}(\bar{L}(a, b), \bar{L}_\Delta(a, b)) = \frac{A_L}{2} \tanh\left(\frac{a}{2}\right) \sum_{x=0}^{\Delta-1} e^{-a|x-b|}.$$

However as a increases the statistical distance increases. The most direct way to see this is to consider the case for $\Delta = 1$, where the statistical distance reduces to

$$\mathrm{SD}(\bar{L}(a, b), \bar{L}_1(a, b)) = \frac{A_L}{2} \tanh\left(\frac{a}{2}\right).$$

Thus by varying a, within the range of values where the average padding length remains close to b, we can strike different tradeoffs between cover difference and statistical distance. Note that when $\Delta = 1$ the value of the statistical distance is equal to the peak value of the distribution. Thus, from the examples displayed in Figures 5 and 6 we see that for the same average length of 15, and $\Delta = 1$ the Laplace yields a statistical distance of 0.1496, whereas for the uniform distribution its value is 0.0322. The relationship between the peak value and the statistical distance for $\Delta = 1$ holds also for the Gaussian distribution. Thus, due to its flatter peak, a Gaussian distribution appears more promising as it might be able to approach the statistical distance of the uniform distribution more closely. Indeed, this turns out to be the case.

5.4 Discrete Gaussian Distribution

We opt for a discrete form of the Gaussian distribution that is obtained by rounding rather than sampling points from the probability density function. This means that the probability of an integer x is obtained by integrating the PDF over the interval $(x - \frac{1}{2}, x + \frac{1}{2}]$ rather than sampling the PDF at x. As we describe in Section C, a rounded Gaussian is more appealing in terms of implementation. Analogous to the previous setting, we shift the Gaussian by μ and truncate all values less than zero. That is, for any integer value $x \geq 0$ the probability mass function is given by:

$$p_{\bar{N}}(x) = \frac{A_G}{\sigma\sqrt{2\pi}} \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2} dt,$$

where

$$\frac{1}{A_G} = 1 - \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{-\frac{1}{2}} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2} dt.$$

Note that due to truncation the actual mean will be slightly offset from μ, and even if we ignore the effect of truncation, the variance

of the rounded Gaussian is still not equal to σ^2, although it is relatively close. A better approximation is given by $\sigma^2 + \frac{1}{12}$, which can be viewed as an application of the well-known Sheppard's correction [46], or a consequence of Janson's result [23] showing that the variance of the rounded Gaussian approaches this value in the limit as σ increases.

An illustrative example of the Gaussian distribution is shown in Figure 7. We set parameters as before, by setting $\mu = b = \frac{l}{2}$ and vary σ to reach a suitable compromise between cover difference and statistical distance. In the example displayed in Figure 7 we set $\sigma = 5.0$ to obtain a cover difference of 0.0009 when $\Delta = 1$ and 0.0052 when $\Delta = 3$, which is comparable to the respective values of 0.0017 and 0.0069 for the Laplace distribution[3]. Yet the Gaussian peaks at 0.0797, almost half the value of the Laplace distribution (0.1496). Thus, in this case, we observe that the statistical distance when $\Delta = 1$ is better for the Gaussian than the Laplace distribution for comparable values of cover difference. Indeed in our numerical evaluations the Gaussian always struck a better tradeoff between cover difference and statistical distance than the Laplace distribution. Accordingly, we make it our preferred choice and focus on comparing the performance of the uniform distribution to that of the Gaussian distribution.

Clearly a Gaussian can do much better than the uniform distribution when our measure is solely the cover difference, since in the case of the Gaussian it can be made arbitrarily small for the same average padding length. However, as we can observe from the example just described, in the case of a single sample the uniform distribution still outperforms the Gaussian distribution in terms of statistical distance. As the number of samples increases, however, the statistical distance will increase in both cases – but at different rates, as predicated by Theorem 4.3. As we shall observe from our experimental results, the Gaussian catches up and outperforms the uniform distribution already at a relatively small number of samples.

The application of Theorem 4.3 to the uniform distribution is fairly straightforward, but not as much in the case of a rounded Gaussian distribution. This is mainly because without a closed form expression for the cumulative distribution function of a Gaussian we cannot express its KL divergence as a concise expression. Nevertheless we can evaluate it numerically, which was our preferred approach, or we can obtain a fairly good and quick approximation as we now describe. For any offset Δ the cover difference is given by:

$$\begin{aligned}\mathrm{CD}(\bar{N}(\mu,\sigma),\bar{N}_\Delta(\mu,\sigma)) &= \sum_{x=0}^{\Delta-1}\frac{p_{\bar{N}}(x)}{2}\\ &= \frac{A_G}{2\sigma\sqrt{2\pi}}\int_{-\frac{1}{2}}^{\Delta-\frac{1}{2}} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2}dt\,.\end{aligned}$$

As for the KL-divergence, when the cover difference is small we can approximate it by using the formula for KL-divergence between two continuous Gaussian distributions [28]. Namely for two continuous Gaussian distributions $\widetilde{N}(\mu_1,\sigma_1)$ and $\widetilde{N}(\mu_2,\sigma_2)$ the KL divergence is given by:

$$\mathrm{D}(\widetilde{N}(\mu_1,\sigma_1)\,\|\,\widetilde{N}(\mu_2,\sigma_2)) = \ln\left[\frac{\sigma_2}{\sigma_1}\right] + \frac{\sigma_1^2+(\mu_1-\mu_2)^2}{2\sigma_2^2} - \frac{1}{2}\,.$$

Now, letting $\mu_1 - \mu_2 = \Delta$ and $\sigma_1 = \sigma_2 = \sigma$, we obtain

$$\mathrm{D}(\bar{N}(\mu,\sigma)\,\|\,\bar{N}_\Delta(\mu,\sigma)) \approx \frac{\Delta^2}{2\sigma^2}\,.$$

5.5 Multisample Attacks on the Uniform and Gaussian Distributions

Besides yielding a generic upper bound for the distinguishing advantage between two distributions in a multi-user setting, Theorem 4.3 provided the intuition that led us to the truncated discrete Gaussian as a suitable alternative to the uniform distribution. However for these specific distributions we can quantify their security more accurately than the upper bounds resulting from Theorem 4.3. Accordingly, let us now turn our attention to identifying suitable distinguisher strategies and corresponding lower bounds. In Section 4.3 we described a generic attack to derive a lower bound for cover difference in a multisample setting. In that attack the distinguisher's strategy was to hope for the occurrence of at least a single sample that identifies the distribution from which it came from. In such a case, the output of the distinguisher is determined by this single sample. Now in the case of two shifted uniform distributions this turns out to be the best that a distinguisher can do. This is easily verified by noting that all other sample vectors are equally likely to have come from either distribution, and thus do not contribute to the statistical distance. It then follows that

$$\mathrm{SD}(U^q(l),U_\Delta^q(l)) = \mathrm{CD}(U^q(l),U_\Delta^q(l)) = 1-\left(1-\frac{\Delta}{l+1}\right)^q.$$

Now, since the statistical distance is an upper bound on the success probability of any distinguisher and this attack's success probability matches it exactly, the attack is optimal. The corresponding strategy for the distinguisher is therefore to look for a sample that determines the distribution with certainty and if no such sample is present it outputs a bit at random.

In the case of two shifted truncated discrete Gaussians, the cover difference can be made arbitrarily small but the attacker can still gather information from its set of samples even if none of them identifies the distribution from which it originated. One strategy is to determine, for each sample, which distribution it is more likely to have originated from and take a majority vote across all samples. This strategy was considered by Sahai and Vadhan in [44] where it was shown that its success probability can be bounded from below using Hoeffding's inequality. However, as pointed out by Reyzin[41] this bound is only meaningful for relatively large sample sets, namely when $q > 1/\epsilon^2$ where ϵ is the statistical distance for a single sample. Thus the Hoeffding bound is of little use for our setting. In turn, in [41] Reyzin provides an alternative bound that covers small sample sets. His bound, however, is only applicable if at least one of the two distributions has a peak that is less than half the statistical distance between the two distributions. Accordingly this limits its applicability to the case of truncated Gaussians since the statistical distance when $\Delta = 1$ is precisely equal to the peak value of the Gaussian, which violates this condition. Moreover,

[3]These values were computed numerically in Python.

even when this condition is satisfied we do not obtain a meaningful bound for the typical values that we are interested in.

Thus the asymptotic techniques typically employed in cryptography do not seem to work in this case. Instead we will derive an attack strategy and a fairly accurate approximation of its success probability using standard statistical techniques. The attacker's strategy is as follows: if any of the samples identifies the distribution with certainty then that sample determines the output, otherwise the adversary computes the average of the samples and its output is determined by whether the average value exceeds a certain threshold or not. In fact this strategy is known to be optimal when attempting to distinguish between two discrete (sampled not rounded) Gaussians, since it corresponds to the maximum likelihood test and by the Neyman-Pearson lemma it is optimal [33]. In order to approximate the success probability of this attack we compute the means of the two distributions conditioned on the event that none of the samples is unique to any of the two distributions. We set the threshold value to be the midpoint between the means. We now use the fact that the standard deviation of the sample mean for a distribution with standard deviation σ', is $\sigma'/\sqrt{q}$, where q is the sample size, and use this to approximate the distinguishing advantage. That is, we set σ' to be the standard deviation of the rounded Gaussian (using Sheppard's correction) and use the CDF of a standard Gaussian with these parameters to model the distribution of the sample mean to approximate the probability that the sample mean from each distribution is below the threshold. Then our approximation of the distinguishing advantage of this adversary is given by:

$$\Pr[E] + \Pr\left[\overline{E}\right]\left\{\Pr\left[0 \leftarrow \mathcal{A} \mid b = 0, \overline{E}\right] - \Pr\left[0 \leftarrow \mathcal{A} \mid b = 1, \overline{E}\right]\right\}$$

$$= 1 - (1 - \mathrm{CD}(\bar{N}(\mu,\sigma), \bar{N}_\Delta(\mu,\sigma)))^q) +$$

$$(1 - \mathrm{CD}(\bar{N}(\mu,\sigma), \bar{N}_\Delta(\mu,\sigma)))^q) \times$$

$$\left\{\mathrm{CDF}\left(\frac{\bar{\mu}_0 + \bar{\mu}_1}{2}, \bar{\mu}_0, \sqrt{\sigma^2 + \frac{1}{12}}\right) - \mathrm{CDF}\left(\frac{\bar{\mu}_0 + \bar{\mu}_1}{2}, \bar{\mu}_1, \sqrt{\sigma^2 + \frac{1}{12}}\right)\right\}.$$

In the above E denotes the event that one of the samples identifies the distribution with certainty, and the values $\bar{\mu}_0$ and $\bar{\mu}_1$ denote the means of the two distributions conditioned on E not occurring. As we shall see next, our experiments indicate that when the truncated component is small this approximation is fairly accurate. Furthermore, due to the similarity between a rounded discrete Gaussian distribution and a sampled discrete Gaussian distribution, we expect this attack strategy to be close to optimal in practice.

5.6 Comparing Distributions

Figure 8 shows a comparison of the security offered by each of the three padding distributions for a fixed set of parameters, namely $U(100)$, $\bar{L}(0.12, 50)$, and $\bar{N}(50, 15)$. We evaluated the distinguishing advantages for each distribution when $\Delta = 1$. We chose these parameters such that (a) all three yield an average length close to 50, and (b) the Gaussian and Laplace distributions yield a similar cover difference for $\Delta = 1$. With these parameters, the average overhead is 50 for the uniform distribution, 50.02 for the truncated Gaussian distribution, and 50.07 for the truncated Laplace distribution.

The solid black line shows the distinguishing advantage for the best possible attack against the uniform distribution that was described in Section 4.3 and Section 5.5. As noted already, the distinguishing advantage of this attack matches exactly the statistical distance, which for the uniform distribution, also turns out to be equal to the cover difference. Thus the black plot represents not only the adversary's ability to distinguish, but its ability to distinguish with *certainty*. In contrast the orange and teal solid lines represents the distinguishing advantage of the threshold attack based on the sample mean against the truncated Gaussian and truncated Laplace distributions respectively. The solid-line plots were computed using our approximation formula derived at the end of Section 5.4 and its adaptation to the Laplace distribution, by replacing $\bar{N}(m, \sigma)$ with $\bar{L}(a, b)$ and the variance σ^2 with $2a^{-2}$. The matching black crosses, '×' and '+', were evaluated by simulating the attack and counting the number of times the attack output 1 when presented with samples from each distribution—corresponding to the D-HIDE advantage in Section 3.4. Each point was evaluated by simulating the attack 100,000 times for each of the two distributions and then using these values to calculate the distinguishing advantage. The dotted plots show upper bounds for the uniform and truncated Gaussian distributions obtained via Theorem 4.3, and the dash-dotted plots at the bottom represent the cover difference for the Gaussian and Laplace distributions.

For the Laplace and Gaussian distributions one can vary a and σ to strike different trade-offs between statistical distance and cover difference. However, in our experiments, the Gaussian distribution emerged as the favoured choice as it always outperformed the Laplace distribution. When comparing the solid black and solid orange plots, we observe that significantly more samples are required for the truncated Gaussian distribution than the uniform distribution to attain the same distinguishing advantage. Thus already when considering plain distinguishing advantage (equivalent to statistical distance for both distributions) the truncated Gaussian performs significantly better than the uniform distribution. When considering the adversary's ability to distinguish with certainty (CD), the performance disparity is much more pronounced—the teal dash-dotted line vs the solid black line. Thus we observe that the proportion of the distinguishing advantage in which the adversary is certain about its output is rather low for the truncated Gaussian and Laplace, whereas for the uniform distribution the distinguishing advantage matches exactly its ability to distinguish with certainty.

6 SUMMARY AND OPEN PROBLEMS

In summary, the rationale leading us to the Gaussian distribution was as follows. The analysis in Section 4 inidicates that distributions with a probability mass concentrated in the middle and small tails are beneficial. Combined with the analogy to Differential Privacy outlined in Section 5.2 this pointed us to two natural candidates, the Laplace and the Gaussian distributions. In our analysis, the Gaussian emerged as the preferred choice due to its lower peak, which translates to a lower single-sample statistical distance. This

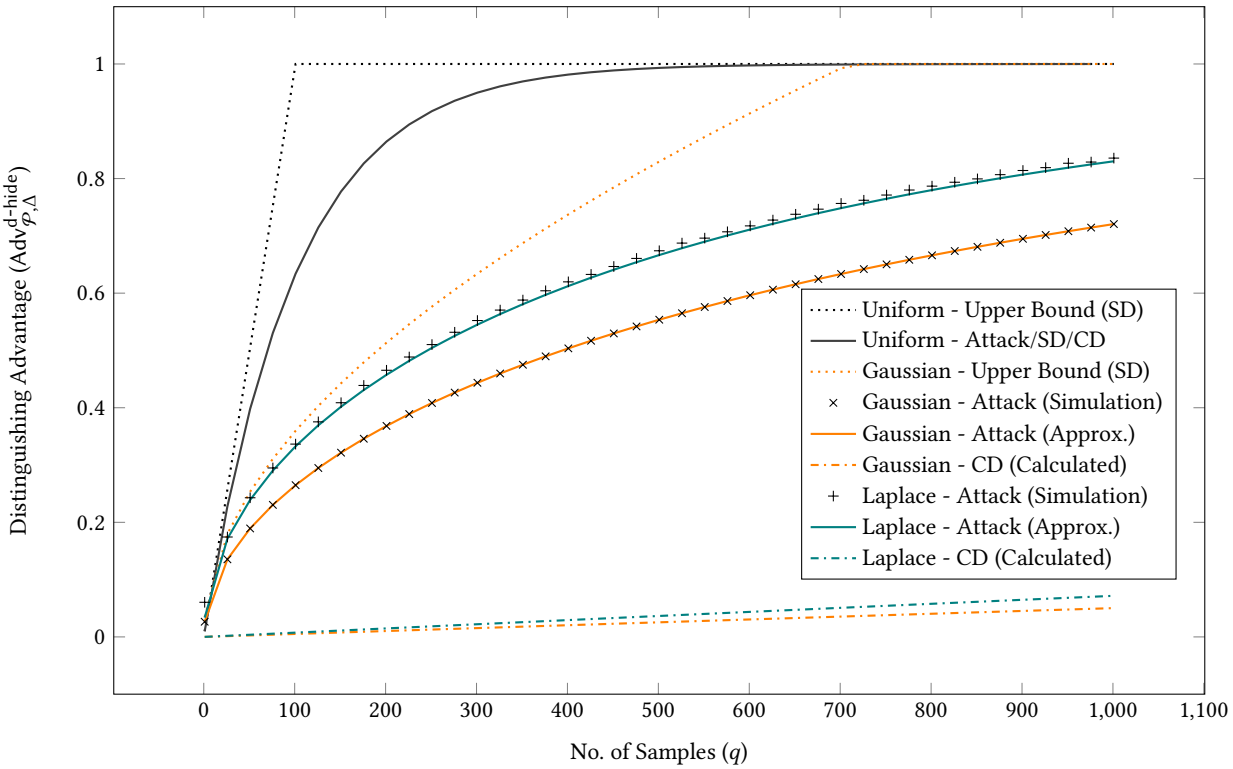

Figure 8: Comparison between Uniform and Gaussian padding, ($\Delta = 1$, average padding length $= 50$).

was reflected in our experiments where the Gaussian always outperformed the Laplace. That said, it remains an open question whether the Gaussian distribution is the optimal choice. Indeed even formulating what one means by a distribution being optimal for this setting is a challenging task, as there are many variables to consider, such as the cover difference, the statistical distance, the average overhead, and the shift Δ. It may well be that no single distribution is optimal over the complete range of possible values. We did not pursue this direction in this work, and we leave it as an open problem.

Nevertheless, we offer the following informal argument in support of Gaussian padding as a suitable choice. The central limit theorem says that the average of multiple samples (from any distribution) will quickly approach a Gaussian distribution. Thus the averaging attack described in Section 5.5, which turns out to be optimal for Gaussian padding (by the Neyman-Pearson lemma), is, in fact, applicable to all padding distributions once a relatively small amount of samples is reached. On the other hand, other distributions may also be susceptible to more severe attacks, as is the case for uniform padding. We, therefore, think it is unlikely that another distribution can significantly outperform Gaussian padding.

In Appendix B we compare empirically how the three types of padding affect attacks such as CRIME and BREACH that exploit the use of compression in combination with TLS. Here we note that Gaussian padding increases the effort required by these attacks substantially already at a reasonable level of overhead. The countermeasure to CRIME was to disable compression in TLS. Accordingly, the padding overhead may be compensated or outweighed by the compression if it is enabled. In Appendix C we discuss potential security pitfalls in implementing Gaussian padding and outline how to implement it securely.

ACKNOWLEDGMENTS

We thank Marc Fischlin for helpful discussions in the early stages of this work. We are also grateful to Kenny Paterson and the anonymous CCS reviewers for their constructive comments on earlier drafts of this work.

Jean Paul Degabriele was supported by the German Federal Ministry of Education and Research and the Hessen State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

REFERENCES

[1] Martin R. Albrecht and Kenneth G. Paterson. 2016. Lucky Microseconds: A Timing Attack on Amazon's s2n Implementation of TLS. In *EUROCRYPT 2016, Part I (LNCS, Vol. 9665)*, Marc Fischlin and Jean-Sébastien Coron (Eds.). Springer, Heidelberg, 622–643. https://doi.org/10.1007/978-3-662-49890-3_24

[2] Martin R. Albrecht, Kenneth G. Paterson, and Gaven J. Watson. 2009. Plaintext Recovery Attacks against SSH. In *2009 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 16–26. https://doi.org/10.1109/SP.2009.5

[3] Nadhem J. AlFardan and Kenneth G. Paterson. 2013. Lucky Thirteen: Breaking the TLS and DTLS Record Protocols. In *2013 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 526–540. https://doi.org/10.1109/SP.2013.42

[4] Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Nicky Mouha, and Kan Yasuda. 2014. How to Securely Release Unverified Plaintext in Authenticated Encryption. In *ASIACRYPT 2014, Part I (LNCS, Vol. 8873)*, Palash Sarkar and Tetsu

Iwata (Eds.). Springer, Heidelberg, 105–125. https://doi.org/10.1007/978-3-662-45611-8_6

[5] Gilles Barthe, Sonia Belaïd, Thomas Espitau, Pierre-Alain Fouque, Mélissa Rossi, and Mehdi Tibouchi. 2019. GALACTICS: Gaussian Sampling for Lattice-Based Constant- Time Implementation of Cryptographic Signatures, Revisited. In *ACM CCS 2019*, Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz (Eds.). ACM Press, 2147–2164. https://doi.org/10.1145/3319535.3363223

[6] Guy Barwell, Daniel Page, and Martijn Stam. 2015. Rogue Decryption Failures: Reconciling AE Robustness Notions. In *15th IMA International Conference on Cryptography and Coding (LNCS, Vol. 9496)*, Jens Groth (Ed.). Springer, Heidelberg, 94–111. https://doi.org/10.1007/978-3-319-27239-9_6

[7] Daniel Bleichenbacher. 1998. Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1. In *CRYPTO'98 (LNCS, Vol. 1462)*, Hugo Krawczyk (Ed.). Springer, Heidelberg, 1–12. https://doi.org/10.1007/BFb0055716

[8] Alexandra Boldyreva, Jean Paul Degabriele, Kenneth G. Paterson, and Martijn Stam. 2014. On Symmetric Encryption with Distinguishable Decryption Failures. In *FSE 2013 (LNCS, Vol. 8424)*, Shiho Moriai (Ed.). Springer, Heidelberg, 367–390. https://doi.org/10.1007/978-3-662-43933-3_19

[9] G. E. P. Box and Mervin E. Muller. 1958. A Note on the Generation of Random Normal Deviates. *The Annals of Mathematical Statistics* 29, 2 (1958), 610 – 611. https://doi.org/10.1214/aoms/1177706645

[10] Leon Groot Bruinderink, Andreas Hülsing, Tanja Lange, and Yuval Yarom. 2016. Flush, Gauss, and Reload - A Cache Attack on the BLISS Lattice-Based Signature Scheme. In *CHES 2016 (LNCS, Vol. 9813)*, Benedikt Gierlichs and Axel Y. Poschmann (Eds.). Springer, Heidelberg, 323–345. https://doi.org/10.1007/978-3-662-53140-2_16

[11] Brice Canvel, Alain P. Hiltgen, Serge Vaudenay, and Martin Vuagnoux. 2003. Password Interception in a SSL/TLS Channel. In *CRYPTO 2003 (LNCS, Vol. 2729)*, Dan Boneh (Ed.). Springer, Heidelberg, 583–599. https://doi.org/10.1007/978-3-540-45146-4_34

[12] Benny Chor and Eyal Kushilevitz. 1990. Secret Sharing Over Infinite Domains (Extended Abstract). In *CRYPTO'89 (LNCS, Vol. 435)*, Gilles Brassard (Ed.). Springer, Heidelberg, 299–306. https://doi.org/10.1007/0-387-34805-0_27

[13] Wei Dai, Viet Tung Hoang, and Stefano Tessaro. 2017. Information-Theoretic Indistinguishability via the Chi-Squared Method. In *CRYPTO 2017, Part III (LNCS, Vol. 10403)*, Jonathan Katz and Hovav Shacham (Eds.). Springer, Heidelberg, 497–523. https://doi.org/10.1007/978-3-319-63697-9_17

[14] Jean Paul Degabriele and Marc Fischlin. 2018. Simulatable Channels: Extended Security that is Universally Composable and Easier to Prove. In *ASIACRYPT 2018, Part III (LNCS, Vol. 11274)*, Thomas Peyrin and Steven Galbraith (Eds.). Springer, Heidelberg, 519–550. https://doi.org/10.1007/978-3-030-03332-3_19

[15] Jean Paul Degabriele and Kenneth G. Paterson. 2007. Attacking the IPsec Standards in Encryption-only Configurations. In *2007 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 335–349. https://doi.org/10.1109/SP.2007.8

[16] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *TCC 2006 (LNCS, Vol. 3876)*, Shai Halevi and Tal Rabin (Eds.). Springer, Heidelberg, 265–284. https://doi.org/10.1007/11681878_14

[17] Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science* 9, 3-4 (2014), 211–407.

[18] Kevin P. Dyer, Scott E. Coull, Thomas Ristenpart, and Thomas Shrimpton. 2012. Peek-a-Boo, I Still See You: Why Efficient Traffic Analysis Countermeasures Fail. In *2012 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 332–346. https://doi.org/10.1109/SP.2012.28

[19] Kai Gellert, Tibor Jager, Lin Lyu, and Tom Neuschulten. 2021. On Fingerprinting Attacks and Length-Hiding Encryption. Cryptology ePrint Archive, Report 2020/824. https://eprint.iacr.org/2021/1027.

[20] Yoel Gluck, Neal Harris, and Angelo Prado. 2013. BREACH: Reviving the CRIME attack, Vol. 2013. Black Hat USA. http://breachattack.com

[21] Paul Grubbs, Marie-Sarah Lacharité, Brice Minaud, and Kenneth G. Paterson. 2018. Pump up the Volume: Practical Database Reconstruction from Volume Leakage on Range Queries. In *ACM CCS 2018*, David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang (Eds.). ACM Press, 315–331. https://doi.org/10.1145/3243734.3243864

[22] Andreas Hülsing, Tanja Lange, and Kit Smeets. 2018. Rounded Gaussians - Fast and Secure Constant-Time Sampling for Lattice-Based Crypto. In *PKC 2018, Part II (LNCS, Vol. 10770)*, Michel Abdalla and Ricardo Dahab (Eds.). Springer, Heidelberg, 728–757. https://doi.org/10.1007/978-3-319-76581-5_25

[23] Svante Janson. 2006. Rounding of continuous random variables and oscillatory asymptotics. *Ann. Probab.* 34, 5 (09 2006), 1807–1826. https://doi.org/10.1214/009117906000000232

[24] Angshuman Karmakar, Sujoy Sinha Roy, Oscar Reparaz, Frederik Vercauteren, and Ingrid Verbauwhede. 2018. Constant-Time Discrete Gaussian Sampling. *IEEE Trans. Comput.* 67, 11 (2018), 1561–1571. https://doi.org/10.1109/TC.2018.2814587

[25] Georgios Kellaris, George Kollios, Kobbi Nissim, and Adam O'Neill. 2016. Generic Attacks on Secure Outsourced Databases. In *ACM CCS 2016*, Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi (Eds.). ACM Press, 1329–1340. https://doi.org/10.1145/2976749.2978386

[26] John Kelsey. 2002. Compression and Information Leakage of Plaintext. In *FSE 2002 (LNCS, Vol. 2365)*, Joan Daemen and Vincent Rijmen (Eds.). Springer, Heidelberg, 263–276. https://doi.org/10.1007/3-540-45661-9_21

[27] S. Kent. 2005. *IP Encapsulating Security Payload (ESP)*. RFC 4303. IETF. http://tools.ietf.org/rfc/rfc4303.txt

[28] Solomon Kullback. 1959. *Information Theory and Statistics*. Wiley, New York.

[29] Marc Liberatore and Brian Neil Levine. 2006. Inferring the source of encrypted HTTP connections. In *ACM CCS 2006*, Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati (Eds.). ACM Press, 255–263. https://doi.org/10.1145/1180405.1180437

[30] David A. McGrew and John Viega. 2004. The Security and Performance of the Galois/Counter Mode (GCM) of Operation. In *INDOCRYPT 2004 (LNCS, Vol. 3348)*, Anne Canteaut and Kapalee Viswanathan (Eds.). Springer, Heidelberg, 343–355.

[31] Christopher Meyer, Juraj Somorovsky, Eugen Weiss, Jörg Schwenk, Sebastian Schinzel, and Erik Tews. 2014. Revisiting SSL/TLS Implementations: New Bleichenbacher Side Channels and Attacks. In *USENIX Security 2014*, Kevin Fu and Jaeyeon Jung (Eds.). USENIX Association, 733–748.

[32] Daniele Micciancio and Michael Walter. 2017. Gaussian Sampling over the Integers: Efficient, Generic, Constant-Time. In *CRYPTO 2017, Part II (LNCS, Vol. 10402)*, Jonathan Katz and Hovav Shacham (Eds.). Springer, Heidelberg, 455–485. https://doi.org/10.1007/978-3-319-63715-0_16

[33] Jerzy Neyman and Egon Sharpe Pearson. 1933. IX. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 231, 694-706 (1933), 289–337.

[34] Y. Nir and A. Langley. 2018. *ChaCha20 and Poly1305 for IETF Protocols*. RFC 8439. IETF. http://tools.ietf.org/rfc/rfc8439.txt

[35] Kenneth G. Paterson and Nadhem J. AlFardan. 2012. Plaintext-Recovery Attacks Against Datagram TLS. In *NDSS 2012*. The Internet Society.

[36] Kenneth G. Paterson, Thomas Ristenpart, and Thomas Shrimpton. 2011. Tag Size Does Matter: Attacks and Proofs for the TLS Record Protocol. In *ASIACRYPT 2011 (LNCS, Vol. 7073)*, Dong Hoon Lee and Xiaoyun Wang (Eds.). Springer, Heidelberg, 372–389. https://doi.org/10.1007/978-3-642-25385-0_20

[37] Raphael C. W. Phan and Serge Vaudenay. 2009. On the Impossibility of Strong Encryption Over $\aleph_0$. In *Coding and Cryptology*, Yeow Meng Chee, Chao Li, San Ling, Huaxiong Wang, and Chaoping Xing (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 202–218.

[38] Alfredo Pironti, Pierre-Yves Strub, and Karthikeyan Bhargavan. 2012. *Identifying Website Users by TLS Traffic Analysis: New Attacks and Effective Countermeasures*. Research Report RR-8067. INRIA. INRIA Technical Report RR-8067.

[39] Anup Rao and Amir Yehudayoff. 2020. *Communication Complexity: and Applications*. Cambridge University Press.

[40] Renato Renner. 2005. On the variational distance of independently repeated experiments. *CoRR* abs/cs/0509013 (2005). arXiv:cs/0509013 http://arxiv.org/abs/cs/0509013

[41] Leonid Reyzin. 2004. *A note on the statistical difference of small direct products*. Technical Report. Boston University Computer Science Department.

[42] Juliano Rizzo and Thai Duong. 2012. The CRIME attack. In *ekoparty security conference*, Vol. 2012.

[43] Phillip Rogaway. 2002. Authenticated-Encryption With Associated-Data. In *ACM CCS 2002*, Vijayalakshmi Atluri (Ed.). ACM Press, 98–107. https://doi.org/10.1145/586110.586125

[44] Amit Sahai and Salil P. Vadhan. 1997. Manipulating statistical difference. In *Randomization Methods in Algorithm Design, Proceedings of a DIMACS Workshop, Princeton, New Jersey, USA, December 12-14, 1997 (DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 43)*, Panos M. Pardalos, Sanguthevar Rajasekaran, and José Rolim (Eds.). DIMACS/AMS, 251–270. http://dimacs.rutgers.edu/Volumes/Vol43.html

[45] Roei Schuster, Vitaly Shmatikov, and Eran Tromer. 2017. Beauty and the Burst: Remote Identification of Encrypted Video Streams. In *USENIX Security 2017*, Engin Kirda and Thomas Ristenpart (Eds.). USENIX Association, 1357–1374.

[46] W. F. Sheppard. 1897. On the Calculation of the most Probable Values of Frequency-Constants, for Data arranged according to Equidistant Division of a Scale. *Proceedings of the London Mathematical Society* s1-29, 1 (11 1897), 353–380. https://doi.org/10.1112/plms/s1-29.1.353 arXiv:https://academic.oup.com/plms/article-pdf/s1-29/1/353/4407416/s1-29-1-353.pdf

[47] Cihangir Tezcan and Serge Vaudenay. 2011. On Hiding a Plaintext Length by Preencryption. In *ACNS 11 (LNCS, Vol. 6715)*, Javier Lopez and Gene Tsudik (Eds.). Springer, Heidelberg, 345–358. https://doi.org/10.1007/978-3-642-21554-4_20

[48] M. Thomson. 2019. *Example Handshake Traces for TLS 1.3*. RFC 8448. IETF. http://tools.ietf.org/rfc/rfc8448.txt

[49] Martin Thomson and Sean Turner. 2021. Using TLS to Secure QUIC – draft-ietf-quic-tls-34. https://tools.ietf.org/html/draft-ietf-quic-tls-34.

[50] Serge Vaudenay. 2002. Security Flaws Induced by CBC Padding - Applications to SSL, IPSEC, WTLS.... In *EUROCRYPT 2002 (LNCS, Vol. 2332)*, Lars R. Knudsen (Ed.). Springer, Heidelberg, 534–546. https://doi.org/10.1007/3-540-46035-7_35

[51] Andrew M. White, Austin R. Matthews, Kevin Z. Snow, and Fabian Monrose. 2011. Phonotactic Reconstruction of Encrypted VoIP Conversations: Hookt on Fon-iks. In *2011 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 3–18. https://doi.org/10.1109/SP.2011.34

[52] T. Ylonen and C. Lonvick. 2006. *The Secure Shell (SSH) Transport Layer Protocol*. RFC 4253. IETF. http://tools.ietf.org/rfc/rfc4253.txt

A PRIOR TREATMENTS OF LENGTH HIDING

We provide here an overview of PRS11 and TV11 and discuss their respective limitations.

A.1 Paterson-Ristenpart-Shrimpton 2011

In their analysis of TLS 1.2, Paterson, Ristenpart, and Shrimpton introduced a security notion called Length-Hiding Authenticated Encryption (LHAE) [36]. They consider encryption with an extended interface which additionally takes as input a value ℓ specifying the desired ciphertext length. The encryption algorithm will in turn either return a ciphertext of the specified length or the special symbol $\perp$ if it cannot accommodate this length. Then the LHAE game is defined analogously to the AE game with a left-or-right encryption oracle taking inputs of the form (ℓ, m_0, m_1). However in the LHAE game, it is no longer required that $|m_0| = |m_1|$, instead an output is returned as long as $\mathcal{E}_K(\ell, m_0) \neq \perp$ and $\mathcal{E}_K(\ell, m_1) \neq \perp$. We refer to [36] for a complete description of the LHAE game.

This security definition was motivated by the use of padding in the TLS protocol and an attack against TLS 1.2, discovered by the same authors, that exploited this padding. In essence, for some TLS ciphertext encrypted using CBC and containing non-minimal padding, an attacker can turn it into a distinct but valid ciphertext by truncating the last block and flipping some bits in the IV. This is an infringement of ciphertext integrity thereby rendering the scheme insecure in the AE sense. On the other hand, it is unclear whether this attack is a real nuisance to the security of TLS. In particular, it does not result in a plaintext forgery, as the plaintext is unaltered, and does not seem to undermine the confidentiality of TLS either, since the adversary will not be able to observe any difference in behaviour on the TLS connection for the forged ciphertext. Nevertheless, a simple variation of the above attack can be used to distinguish two equal-length ciphertexts containing messages of differing lengths, say an encryption of "YES" from an encryption of "NO". Specifically, if the original ciphertext contains a message of the appropriate size then the forged ciphertext will be valid, but will otherwise result in an invalid ciphertext and generate a TLS error that is noticeable to the adversary. This is certainly a practical concern that TLS aims to protect against through its use of padding. The authors of [36] then argue that by allowing the adversary to be challenged on messages of differing lengths, the LHAE definition clearly captures this latter attack whereas the AE definition does not.

Despite the above motivation, in our view, the LHAE definition has some undesirable features, and, in addition, it does not capture all aspects of length-hiding encryption that one would expect. Intuitively, the name suggests that any scheme meeting this notion would hide the lengths of its messages to some extent. On the contrary, however, any AE-secure scheme with constant ciphertext expansion, which trivially leaks the full message length, will meet this notion. Thus, if an encryption scheme meets this notion there is no guarantee whatsoever that it hides the message-length in any way. This may seem at odds with the motivation just described, where LHAE appears to capture a broader class of attacks than AE. The reason for this is a circularity in the security definition, where the strength of the LHAE notion depends on the scheme itself. Specifically, the wider the range of ciphertext lengths that the scheme admits for a given message length, the stronger the security notion. This dependency is undesirable as it does not allow for an objective comparison between distinct encryption schemes, as the notion yields different security properties for different schemes.

Additionally, it turns out that the LHAE definition is not significantly stronger than standard AE security. This is subject to whether AE security is formulated using left-or-right indistinguishability (as in [36]) or indistinguishability from random bits [43]. To make this comparison we need to extend these two notions for encryption schemes with variable output length, but this is fairly straightforward. It can then be shown that while LHAE is strictly stronger than the left-or-right formulation, any scheme satisfying the indistinguishability from random bits formulation of AE will automatically satisfy LHAE.

Yet, perhaps the most important limitation of the LHAE definition is that it does not provide us with any insight as to how we should choose the ciphertext length. In the LHAE game this is handled by the adversary but in practice the sender must have some procedure for determining the ciphertext length based on the message length. In fact, the overall length-hiding ability of the encryption scheme depends crucially on this procedure. However, in the LHAE defintion, this mechanism is taken completely out of the picture, and thereby, its efficacy cannot be evaluated. Furthermore the LHAE games focusses on a very limited scenario, where the adversary is only challenged on ciphertexts of the same length, and consequently the definition says nothing about an adversary's ability to distinguish ciphertexts of different lengths. For schemes employing randomised procedures to determine the ciphertext length, the probability that two messages map to ciphertexts of differing lengths is rather high. As such the LHAE definition misses many challenge pairs that arise in practice when ciphertext lengths are determined probabilistically. Put differently, an adversary's advantage in the LHAE game does not accurately reflect its advantage in practice, since the probability that the two messages map to ciphertexts of the same length may be quite small.

A.2 Tezcan-Vaudenay 2011

Tezcan and Vaudenay take a very different approach in analysing the security of length-hiding encryption [47]. They focus on the common tactic of appending randomised padding to the message before encryption, a process which they refer to as *preencryption*. They consider a security notion Δ-IND-OTE for encryption, analogous to IND-CPA, with the vital distinction that message lengths are allowed to differ by at most Δ bits. However, in Δ-IND-OTE the adversary can query the left-or-right encryption oracle only once.

They show that appending padding of uniformly distributed length is a nearly optimal preencryption scheme for achieving Δ-IND-OTE security. Roughly, they show that for any preencryption scheme with maximum padding length B, there always exists an adversary with distinguishing advantage $1/(2\lceil \frac{B}{\Delta} \rceil)$. On the other hand, the advantage is bounded above by the statistical distance

Attack Performance vs Padding Overhead				
Overhead	0 bytes	50 bytes	100 bytes	200 bytes
Uniform				
Queries	192	19,367	38,569	76,883
Succ. Prob.	1.0	1.0	1.0	1.0
Gaussian				
Queries	192	576,000	2,304,000	7,680,000
Succ. Prob.	1.0	0.0026	0.0058	0.0026
Laplace				
Queries	192	576,000	2,304,000	7,680,000
Succ. Prob.	1.0	0.2116	0.2714	0.673

Figure 9: Comparing the performance of different padding distributions on the of the CRIME attack.

between the two message length distributions. When the padding length is sampled uniformly over the interval $(0, B]$ and the two messages are Δ bits apart in length, the statistical distance works out to be $\Delta/2B$. Thus when Δ divides B, uniform padding is optimal among all preencryption schemes of maximum expansion B. Arguably, a more general treatment would consider the optimum over the class of preencryption schemes with some fixed *average* expansion. Indeed their attack can be extended to this more general setting, yielding a less optimal lower bound of $1/(4\lceil\frac{2B}{\Delta}\rceil)$ for padding limited to an average expansion of $B/2$.

When compared to [36], the security model of Tezcan and Vaudenay allows one to evaluate and compare the efficacy of preencryption schemes, which [36] does not. However, it overlooks the fact that the size of the padding may leak during decryption, as is the case in the attack described in [36]. Another significant limitation is the fact that the adversary is allowed only one encryption query. As we show in Sections 4 and 5, this turns out to be a critical factor for the near optimality of uniform padding, as it no longer holds when this restriction is lifted.

B APPLICATION TO THE CRIME ATTACK

Until now, we have focused solely on statistical distance as our primary security measure. To get a better sense of what this means in practice, we now consider the effect of length-hiding padding on the well-known CRIME attack [42]. CRIME and other variants like BREACH [20] are chosen plaintext attacks that exploit compression in TLS or higher-layer applications to recover some secret—typically a secure cookie. The attack assumes a malicious script running on the victim's machine that can make HTTP requests to a TLS-protected website, including the secure cookie. Note, however, that the script does not have direct access to the cookie and thus cannot see its value. The attack further assumes that the adversary is able to sniff the victim's encrypted HTTP requests and observe their lengths. The BREACH attack acts similarly but instead targets the HTTP responses returned by the server and their lengths.

The compression algorithm that enables the attack is the ubiquitous DEFLATE algorithm which combines Lempel-Ziv and Huffman coding. The Lempel-Ziv coding in DEFLATE allows for compressing a message containing multiple instances of the same string within that message. CRIME and BREACH exploit the Lempel-Ziv compression in DEFLATE to guess the secret cookie one byte at a time. Namely, the encrypted HTTP message will contain a string such as '`cookie=I+ldQbtvMmfJn146zhRX`', and the malicious script will inject a second string of the form '`cookie=?`', where ? represents a guess of the first cookie byte. An incorrect guess means that only 7 characters will be matched, but a correct guess will match 8 characters, resulting in a compressed message that is one byte shorter. If an encryption scheme like Galois Counter Mode (GCM) is used, the one-byte difference in the message lengths will be propagated to the ciphertext lengths. On the other hand, a block-aligned scheme like CBC encryption does not help either, since the malicious script can pad the message so that the extra byte falls on a block boundary and the adversary would still be able to detect the single-byte compression. Once the first character is guessed correctly, the attacker injects the string '`cookie=I?`' to guess the second character and proceed this way until the full secret is recovered. Assuming a base64 32-character long cookie, the above attack would require around 1024 queries on average. Rizzo and Duong describe how to reduce this by employing a binary search where a single message includes multiple guesses. This way, each character requires only 6 queries, and the whole secret can be recovered in 192 queries.

Length-hiding padding can be used to substantially increase the number of queries required to mount the attack. How much overhead one is willing to accept is very specific to the application at hand. However, it is conceivable that once the number of required queries reaches a sufficiently large value, say in the millions, the attack would be detected or prevented by rate-limiting countermeasures that are already in place to prevent denial-of-service attacks. We have simulated this attack in a proof-of-concept implementation to measure how each padding distribution affects the success of the attack. Once again, Gaussian padding emerged as the favoured choice, with Laplace performing about two orders of magnitude worse and uniform padding performing significantly worse. Namely, for the attack to require a certain number of queries, uniform padding results in much more bandwidth overhead than the other two alternatives. For Gaussian and Laplace padding, when the cover difference is kept low, the best that the adversary can do is to reduce his chance of making an incorrect guess as low as possible. A single incorrect guess at any of the intermediate steps in the attack is enough to foil the success of the whole attack. As such, the best strategy, in this case, is to repeat each query a fixed number of times, calculate the average to reduce the noise from the padding, and decide how to proceed based on that average value. On the other hand, an attacker can do much better with respect to uniform padding since it can be certain that it has made the right guess at every intermediate step in the attack. For instance, if the padding length ranges from 0 to 50 bytes and the adversary is trying to determine whether the compressed message length is 100 bytes or 99 bytes, a padded message that is 150 bytes long could only have originated from the former case. Similarly, a padded message length of 99 bytes can only originate from the latter case. As such, the adversary can keep on making queries until it observes a padded message length that by itself identifies the unpadded message length with certainty. With this approach, the attack succeeds with probability one, but the required number of queries varies with each run of the attack.

Figure 9 shows the results of our simulations. The overhead represents the average padding length (in bytes) for each distribution.

For the Gaussian and Laplace, we fixed the number of queries and calculated the success probabilities by running the attack 5,000 times and dividing the number of successes by the total number of attempts. For uniform padding, we instead calculated the expected number of queries required by the attack over 5,000 trials. In the case of uniform padding, at every step, the adversary is waiting for exactly one padding length value to emerge. For an overhead of 100, the probability of this padding length being sampled is $1/200$. Since this follows a geometric distribution, the expected number of queries (at each step) to sample this value is 200. The attack requires 192 steps, and thus the theoretically expected value is 192×200, which closely matches our experimental results. As such, the number of expected queries increases linearly with the overhead. On the other hand, we have chosen the number of queries for the Gaussian and Laplace so that the success probability for the Gaussian stayed close to 0.002. We then observe that in this case, the number of queries increases, more or less, quadratically with the overhead. This is in agreement with our observation from Section 5 that their statistical distance increases proportionally to $\sqrt{q}$. From the above calculations, we then observe that for uniform padding to increase the expected number of queries up to 7,680,000 would require an overhead of 20,000 bytes per message. In contrast, the Gaussian distribution requires only 200 bytes!

C IMPLEMENTING GAUSSIAN PADDING

Gaussian padding is not as straightforward to implement as uniform padding, and some caution is necessary to avoid potential pitfalls. That said, the Gaussian distribution plays a prominent role in other areas of cryptography, such as Differential Privacy and Lattice-based cryptography, and consequently, it has received a fair amount of scrutiny. The chief concern when implementing Gaussian padding is to protect against side-channel attacks, i.e., not to leak the size of the padding through timing information. Indeed, cache-timing attacks on the Gaussian-noise-sampling procedure have been used to break implementations of the lattice-based signature scheme BLISS, for example [10]. In our treatment, we have always assumed that the padding size is unknown to the adversary, and the presence of such a side channel would clearly invalidate our analysis.

In response to this, a number of follow-up works have proposed algorithms for sampling Gaussian distributions in constant time [5, 22, 24, 32]. However, it should be noted that in lattice-based cryptography, the discrete form of a Gaussian distribution is typically obtained by measuring the amplitude of the continuous probability density function at integer values. On the other hand in [22] Hüsing *et al.* propose the use of a *rounded* discrete Gaussian as it is relatively easy to implement in constant time. A rounded Gaussian corresponds to sampling from a continuous Gaussian distribution and then rounding the output to the closest integer. In our analysis, we assumed a rounded Gaussian distribution and accordingly, the simple approach described in [22] is particularly well-suited for our setting.

The approach proposed in [22] is based on the Box-Muller transform [9], which we now adapt to our setting. The basic Box-Muller transform, reproducded below in Algorithm 1, takes two uniformly-distributed samples and maps them to two independent samples from a *continous* Gaussian distribution. The samples from $\bar{N}(\mu, \sigma)$ are then given by: $\lfloor z_1 \cdot \sigma \rceil + \mu$ and $\lfloor z_2 \cdot \sigma \rceil + \mu$ and discarding *both* samples if either value is less than zero.

Algorithm 1: Box-Muller Transform

Input : Two random samples u_1 and u_2 from $U(1)$
Output: Two random samples z_1 and z_2 from $N(0, 1)$
$r \leftarrow \sqrt{-2 \ln u_1}$
$\theta \leftarrow 2\pi u_2$
$z_1 \leftarrow r \cos\theta$; $z_2 \leftarrow r \sin\theta$
return (z_1, z_2)