Differentially Private Approximate Pattern Matching

Teresa Anna Steiner
DTU Compute, Technical University of Denmark, Kongens Lyngby, Denmark

Abstract

Differential privacy is the de facto privacy standard in data analysis and widely researched in various application areas. On the other hand, analyzing sequences, or *strings*, is essential to many modern data analysis tasks, and those data often include highly sensitive personal data. While the problem of sanitizing sequential data to protect privacy has received growing attention, there is a surprising lack of theoretical studies of algorithms analyzing sequential data that preserve differential privacy while giving *provable guarantees on the accuracy* of such an algorithm. The goal of this paper is to initiate such a study.

Specifically, in this paper, we consider the k-approximate pattern matching problem under differential privacy, where the goal is to report or count all substrings of a given string S which have a Hamming distance at most k to a pattern P, or decide whether such a substring exists. In our definition of privacy, *individual positions of the string S* are protected. To be able to answer queries under differential privacy, we allow some slack on k, i.e. we allow reporting or counting substrings of S with a distance at most $(1+\gamma)k+\alpha$ to P, for a multiplicative error γ and an additive error α. We analyze which values of α and γ are necessary or sufficient to solve the k-approximate pattern matching problem while satisfying ϵ-differential privacy. Let n denote the length of S. We give

- an ϵ-differentially private algorithm with an additive error of $O(\epsilon^{-1}\log n)$ and no multiplicative error for the existence variant;
- an ϵ-differentially private algorithm with an additive error $O(\epsilon^{-1}\max(k,\log n)\cdot\log n)$ for the counting variant;
- an ϵ-differentially private algorithm with an additive error of $O(\epsilon^{-1}\log n)$ and multiplicative error $O(1)$ for the reporting variant for a special class of patterns.

The error bounds hold with high probability. All of these algorithms return a witness, that is, if there exists a substring of S with distance at most k to P, then the algorithm returns a substring of S with distance at most $(1+\gamma)k+\alpha$ to P.

Further, we complement these results by a lower bound, showing that any algorithm for the existence variant which also returns a witness must have an additive error of $\Omega(\epsilon^{-1}\log n)$ with constant probability.

2012 ACM Subject Classification Security and privacy; Theory of computation → Pattern matching

Keywords and phrases Differential privacy, pattern matching

Digital Object Identifier 10.4230/LIPIcs.ITCS.2024.94

Related Version *Full Version*: https://arxiv.org/abs/2311.07415

Funding *Teresa Anna Steiner*: This work was supported by a research grant (VIL51463) from VILLUM FONDEN.

1 Introduction

Analyzing sequential data is essential to many modern data analysis tasks, including signal processing, route planning, and genetic matching. Since those data can include highly sensitive personal data, the problem of *sanitizing sequential data to protect privacy* while *preserving patterns that occur within these sequences* has received growing attention [11, 9, 34, 24, 6, 1, 2, 3, 20, 19, 7, 5, 10, 18, 37, 21, 33]. The applications considered in these papers range from genetic matching [34] over natural language processing [19, 9] to travel

15th Innovations in Theoretical Computer Science Conference (ITCS 2024).
Editor: Venkatesan Guruswami; Article No. 94; pp. 94:1–94:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

pattern mining [24, 11, 20, 10, 18, 37]. These works partially use differential privacy [34, 24, 6, 9, 19, 7, 5, 10, 18, 37, 21, 33] or other privacy measures [1, 2, 3, 11, 20]. The utilities of the proposed algorithms are shown by extensive experiments. Despite this effort led by practitioners, there is a lack of theoretical studies of algorithms analyzing sequential data that preserve differential privacy while giving *provable guarantees on the accuracy* of such an algorithm. The goal of this paper is to initiate such a study.

Differential privacy is the de facto privacy standard used in modern data analysis [36]. Its definition offers strong privacy guarantees and is due to Dwork et al. [12]. Informally, the definition states that the output distributions of an algorithm should be close on close data sets, i.e., the output should not depend much on any single data point. In more detail, we call two data sets which differ in a single data point *neighbouring.* A randomized algorithm is *ϵ-differentially private*, if for any two neighbouring input data sets, the output distributions of the algorithm differ by at most a factor of e^{ϵ}.

A natural data type to model sequential data is a *string*, which is a sequence of symbols drawn from some predefined alphabet. Strings are used to model any type of text data, as well as genetic data and event series. One of the most fundamental problems in string algorithms is the pattern matching problem: For a string S and a pattern string P, decide if P occurs in S (*existence*), count the occurrences of P in S (*counting*), or report all positions in S where P occurs (*reporting*). The pattern matching problem and its variants have been an active research field for more than 50 years with applications ranching from signal processing over computational biology to information retrieval.

In this work, we begin a theoretical study of *differentially private pattern matching* for strings. Specifically, we study the approximate pattern matching problem and show that combining well-known techniques from differential privacy [15] with modern techniques used by the pattern matching community to solve the approximate pattern matching problem [8] can be used to prove interesting new theoretical upper and lower bounds on the error needed by any differentially private algorithm solving the approximate pattern matching problem. We see this as a proof of concept that the field of *differentially private string algorithms* is a promising direction for future research.

In the following, we describe the problem considered in this work in more detail.

Privacy Model and Motivation

In this paper we focus on protecting *individual positions in the string S*, that is, the pattern matching algorithm should have similar output distributions when matching P in S and T, if S and T differ in few positions. That is, we call two strings S and S' neighbouring, if they differ in one position. This privacy model has also been used for strings by Fichtenberger et al. [16] for the problem of counting all occurrences of any pattern of a given length in a stream. It corresponds to *event-level privacy* for continual observation, i.e., instead of protecting an entire user's data, single events are protected [13]. Since the output of the algorithm has a similar distribution whether any single event happened or not, this can be seen as providing *plausible deniability* of any given event. Thus, this model makes sense in settings where a user cares about single events or outliers in their behaviour being concealed, while still allowing the service to draw conclusions about their general behaviour. For example, the string could be a sequence of locations a person visited, and hiding any single position in that sequence corresponds to hiding whether a person visited any particular location at a given time or not. For another example, the string can be a list of items bought by a customer through an online service, and any single purchase is masked. This definition can still allow trends to be detected, e.g. if a user buys chocolate every day, a differentially private algorithm may reveal that the person buys lots of chocolate; however, if a user buys a single sensitive item, e.g. a pregnancy test, this data is concealed.

Approximate Pattern Matching

Note that we cannot hope to solve the pattern matching problem *exactly* while satisfying this definition of differential privacy: For any pattern P, we can easily find strings S and S' such that P occurs in S, P does not occur in S', and S and S' differ in only one position. Thus, any reasonable solution to the exact pattern matching problem with pattern P should be able to differentiate between S and S', which contradicts the goal of differential privacy.

Therefore we study the *k-approximate pattern matching problem*: For a pattern P of length m, a string S of length $n \geq m$, and a parameter $k \leq m$, we want to find all substrings of length m of S, such that the distance between the substring and P is at most k. This problem has been extensively studied in the non-private setting (recent work includes [29, 38, 17, 8, 4], see also the survey by Navarro [26]) since it captures several applications more fully than exact matching: In many applications, the string and the pattern might suffer some corruption, e.g. mutation in DNA sequences, measurement or transmission errors, or typing errors [26]. In this work, we consider the Hamming distance as distance measure. In order to design algorithms that fulfill differential privacy, we allow some slack on k: We want to find all length-m substrings (given by their starting and ending position in S) of distance at most k to P, but we allow the algorithm to return length-m substrings of distance at most $(1+\gamma)k+\alpha$, for a *multiplicative error* γ and an *additive error* α. We also consider the natural *counting* and *existence* variants of this problem (the formal definitions of these problems are given in Section 2). The goal is to analyze which values of γ and α are possible and necessary to solve the approximate pattern matching problem while preserving ϵ-differential privacy.

Results

First, we note that there is a trivial algorithm with additive error $O(m)$, which is ϵ-differentially private for all ϵ: We simply output all substrings of S, i.e. all pairs $(i, i+m-1)$ for $i \leq n-m$. Since this is independent of the string S, the algorithm is differentially private by default, and since the true distance is always a value between 0 and m, the additive error is at most m.

In this paper, we give new trade-offs for the existence, counting and reporting variants of the problem. First, we give an algorithm for the existence variant achieving $O(\log n)$ additive error and no multiplicative error. Then, for counting and reporting, we use results on (non-private) approximate pattern matching [8] to differentiate between patterns fulfilling different properties: If the pattern is close to a periodic string with a small enough period, we can exploit that to give an algorithm for the reporting variant of the approximate pattern matching problem with constant multiplicative error and $O(\log n)$ additive error. Otherwise, we can use the results in [8] to bound the number of substrings in S which can be close to P, and use that fact to give an algorithm for the counting variant. Our upper bound results are summarized in the following two theorems.

▶ **Theorem 1** (Summary of Lemma 15, Theorem 18, Theorem 19, and Lemma 20)**.** *Let n denote the length of input string S, $m \leq n$ the length of pattern P, and $k \leq m$ an integer.*

1. *There exists an algorithm for the existence variant of the k-approximate pattern matching problem which with probability $1-\beta$ has an additive error of at most $\alpha = O(\log(n/\beta)/\epsilon)$ and a multiplicative error $\gamma = 0$.*
2. *For $k = \Omega(\log(n/\beta)/\epsilon)$, there exists an algorithm for the counting variant of the k-approximate pattern matching problem which with probability $1-\beta$ has a multiplicative error of at most $\gamma = O(\log(n/\beta)/\epsilon)$ and an additive error $\alpha = 0$.*

3. *For $k = O(\log(n/\beta)/\epsilon)$, there exists an algorithm for the counting variant of the k-approximate pattern matching problem which with probability $1-\beta$ has an additive error of at most $\alpha = O(\log^2(n/\beta)/\epsilon^2)$ and a multiplicative error $\gamma = 0$.*

Further, all of these algorithms return a witness*, i.e., a length-m substring of S with Hamming distance at most $(1+\gamma)k + \alpha$ to P.*

▶ **Theorem 2** (Informal version of Theorem 18)**.** *Let P be a string of length m. If P has Hamming distance at most $2k$ from a periodic string of period at most $qm/((\log(n/\beta)/\epsilon)+k)$, for some suitable constant q, then there exists an algorithm for the reporting variant of the k-approximate pattern matching problem for pattern P and any string S of length n which with probability $1-\beta$ has a multiplicative error of $O(1)$ and an additive error of $O(\log(n/\beta)/\epsilon)$.*

We complement these results with lower bounds on the necessary additive error for the k-approximate pattern matching problem under ϵ-differential privacy. These lower bounds specifically show that the additive error for the existence variant from Theorem 1 is asymptotically optimal for $m \ll n$:

▶ **Theorem 3** (Informal version of Theorem 21)**.** *Let P be any string of length m and let $k < m$ be an integer. Assume there is an ϵ-differentially private algorithm which solves the existence variant of the k-approximate pattern matching problem for pattern P and any string S and returns a witness, with an additive error at most α with constant probability. Then either $\alpha = \Omega(m-k)$, or both $m = \Omega(\epsilon^{-1}\log n)$ and $\alpha = \Omega(\epsilon^{-1}\log(n/m))$.*

Note that Theorem 3 gives a lower bound that holds for *any* pattern P, no matter if it is close to a periodic substring of small period, or not.

In this work, we mostly care about the privacy-to-accuracy trade-off of the problem. However, for completeness, we show in Appendix A, that the algorithms achieving the upper bounds stated above run in time $O(nm + m^3)$, assuming that any needed random noise can be drawn in constant time. We did not try to optimize this run time.

Related Work

Fichtenberger et al. [16] show how to count all patterns of a bounded length over a stream while preserving differential privacy. It is given as a direct application of their general differentially private counting algorithm. Their privacy model is the same as ours, however, their error definition is an error on the *value of the count*, instead of an error on the Hamming distance, as in our paper.

There is a large body of work on mining frequent patterns or q-grams (substrings of length q) from a set of strings while satisfying differential privacy [5, 9, 6, 24, 7, 19, 10, 37, 21]. In those works, the input data set consists of multiple strings, and two neighbouring data sets differ in one string in the set. The utilities of these algorithm are evaluated by experiments.

There is a line of work on *combinatorial string sanitization* focusing on hiding a *given set of sensitive patterns* [1, 2, 3]. Ajala et al. [1] consider sanitizing the string by replacing letters. They show that the problem of finding the minimum number of letters to be replaced is NP-hard and propose an algorithm. Bernardini et al. [2] propose an algorithm for finding the minimal length string maintaining the order and frequency of all non-sensitive patterns, and another algorithm for finding a string maintaining the order and frequency of all non-sensitive patterns while minimizing the edit distance between the original string and the output string. Bernardini et al. [3] study the connection between string sanitization and frequent pattern mining. Compared to our work, they mask *all* occurrences of sensitive patterns, however, the specific patterns have to be given in advance. On the other hand, our definition hides any single (or any set of few) occurrences of *any* potentially sensitive pattern. Note that in those works, the goal is to mask *exact* occurrences of the sensitive patterns, i.e. it still allows occurrences of substrings which are close to a sensitive pattern.

There is previous work on private pattern matching from a cryptographic perspective with applications in genetic matching [22, 23, 31, 27, 28, 30, 35]: In the model considered in these works, data is held by one party (or the cloud) and queries are sent by another (or multiple other) parties; encryption is used to ensure privacy of the data and the query. In these works, the query party can find out whether their query pattern occurs in the string or collection of strings in the data, while nothing else about the data is revealed to the query party and the query is not revealed to the data holder. In a similar model, two parties each hold a string and want to compare how similar they are, without revealing anything else to each other [32]. Note that the goal in differential privacy is orthogonal to these privacy definitions: In our definition, the data holder knows everything; however, the *query answer* should conceal any individual string positions of the data holder's string.

Paper Organization

The rest of the paper is organized as follows. In Section 2, we formally define the problem and recall some definitions and theorems for differential privacy and strings. In Section 3, we prove Theorems 1 and 2. In Section 4, we prove Theorem 3. Finally, we conclude with some directions for future research (Section 5). In Appendix A, we analyze the runtime of our algorithms.

2 Preliminaries

We denote an interval of integers $\{a, a+1, \dots, b\}$ as $[a, b]$.

2.1 String Preliminaries

A *string* S of length n is a sequence $S[0]S[1]\dots S[n-1]$ of symbols from an alphabet Σ. The length of S is denoted $|S|$. We call $S[a,b] := S[a]S[a+1]\dots S[b]$ a *substring* of S. We denote by $S^{\text{rev}} := S[n-1]S[n-2]\dots S[0]$ the *reverse* of string S. For $k \in \mathbb{N} \cup \{\infty\}$ we denote by S^k the string obtained by concatenating S k times. A string S is called primitive if there does not exist a string T such that $S = T^k$ for $k \geq 2$.

A *period* of a string S is a number $\pi \in [0, n-1]$ such that $S[i] = S[i+\pi]$ for all $i \in [0, n-1-\pi]$. A string S is *periodic* if it has a period π with $\pi < n/2$.

The *Hamming distance* between two strings S and T with $n = |T| = |S|$ is defined as

$$\text{dist}_H(T, S) = |\{i \in [0, n-1] : T[i] \neq S[i]\}|.$$

For a string P of length m and a string S of length n with $n \geq m$, $i \in [0, n-m]$, we call $S[i, i+m-1]$ a *k-mismatch occurrence* if $\text{dist}_H(S[i, i+m-1], P) \leq k$.

2.2 Privacy Definition and Problem Definitions

Two strings S and S' of length n are defined as *neighbouring*, if their Hamming distance is one, i.e., if they differ in one position.

We generally define a *pattern matching algorithm* to be an algorithm taking as input a string S of length n and a pattern P, and outputting either a Boolean value (*existence*), a natural number in $[0, n-1]$ (*counting*), or a subset of $[0, n-1]$ (*reporting*).

We say a pattern matching algorithm $\mathrm{Alg} : \Sigma^* \times \Sigma^m \to \mathrm{range}(\mathrm{Alg})$ is $\epsilon-$*differentially private*, if for all $\mathrm{Out} \subseteq \mathrm{range}(\mathrm{Alg})$, all patterns P of length m and all pairs of neighbouring strings S and S',

$$\Pr(\mathrm{Alg}(S, P) \in \mathrm{Out}) \leq e^\epsilon \cdot \Pr(\mathrm{Alg}(S', P) \in \mathrm{Out}),$$

where the probabilities are taken over the internal randomness of Alg.

▶ **Definition 4** (k-approximate pattern matching problem with one-sided error, reporting variant)**.** *Given a string S of length n, a pattern P of length m and a parameter k, output a set of indices $I \in [0, n-m]$ such that*

1. *If* $\mathrm{dist}_H(P, S[i, i+m-1]) \leq k$ *for an* $i \in [0, n-m]$*, then* $i \in I$*,*
2. *If* $i \in I$ *then* $\mathrm{dist}_H(P, S[i, i+m-1]) \leq (1+\gamma)k + \alpha$*.*

We call γ the multiplicative error *and α the* additive error*.*

In the following, denote by $c_x(S, P)$ the number of positions i in S such that $\mathrm{dist}_H(P, S[i, i+m-1]) \leq x$. If P is clear from context, we will sometimes write $c_x(S)$ for $c_x(S, P)$.

▶ **Definition 5** (k-approximate pattern matching problem with one-sided error, counting variant)**.** *Given a string S of length n, a pattern P of length m and a parameter k, output a number c such that*

1. $c \geq c_k(S, P)$*,*
2. $c \leq c_{(1+\gamma)k+\alpha}(S, P)$*.*

Further, if $c > 0$, additionally output a position i fulfilling $\mathrm{dist}_H(P, S[i, i+m-1]) \leq (1+\gamma)k + \alpha$*. We call i a* witness*. We call γ the* multiplicative error *and α the* additive error*.*

▶ **Definition 6** (k-approximate pattern matching problem with one-sided error, existence variant)**.** *Given a string S of length n, a pattern P of length m and a parameter k, output*

1. *YES, if there exists $i \in [0, n-m]$ such that* $\mathrm{dist}_H(P, S[i, i+m-1]) \leq k$*,*
2. *NO, if there does not exist $i \in [0, n-m]$ such that* $\mathrm{dist}_H(P, S[i, i+m-1]) \leq (1+\gamma)k + \alpha$*.*

Further, if the answer is YES, additionally output a position i fulfilling $\mathrm{dist}_H(P, S[i, i+m-1]) \leq (1+\gamma)k + \alpha$*. We call i a* witness*. We call γ the* multiplicative error *and α the* additive error*.*

2.3 Privacy Preliminaries

First, we collect some definitions to introduce the *Laplace mechanism.*

▶ **Definition 7** (L_1-sensitivity)**.** *Let f be a function $f : \chi \to \mathbb{R}^k$ for some universe χ. The* L_1-sensitivity of f *is defined as*

$$\max_{x,y \text{ neighboring}} ||f(x) - f(y)||_1. \tag{1}$$

▶ **Definition 8.** *The* Laplace distribution *centered at* 0 *with scale b is the distribution with probability density function*

$$f_{\mathrm{Lap}(b)}(x) = \frac{1}{2b} \exp\left(\frac{-|x|}{b}\right).$$

We use $X \sim \mathrm{Lap}(b)$ or just $\mathrm{Lap}(b)$ to denote a random variable X distributed according to $f_{\mathrm{Lap}(b)}(x)$.

▶ **Lemma 9** (Theorem 3.6 in [15]: Laplace Mechanism). *Let f be any function $f : \chi \to \mathbb{R}^k$ with L_1-sensitivity Δ_1. Let $Y_i \sim \mathrm{Lap}(\Delta_1/\epsilon)$ for $i \in [k]$. The mechanism defined as:*

$$A(x) = f(x) + (Y_1, \ldots, Y_k)$$

satisfies ϵ-differential privacy.

The following fact follows directly from the definition of differential privacy, and extends the privacy definition from neighbouring input strings to inputs which have small distance from each other.

▶ **Lemma 10** (Group Privacy for Pattern Matching). *Let S and S' have a Hamming distance at most ℓ, i.e. $\mathrm{dist}_H(S, S') \leq \ell$. Let Alg be an ϵ-differentially private pattern matching algorithm. Then for any pattern P,*

$$\Pr(\mathrm{Alg}(S, P) \in \mathrm{Out}) \leq e^{\ell\epsilon} \cdot \Pr(\mathrm{Alg}(S', P) \in \mathrm{Out}).$$

The following is a well-known Fact which follows immediately from the definition of differential privacy.

▶ **Lemma 11** (Composition Theorem). *Let $\mathrm{Alg}_1 : \chi \to \mathrm{range}(\mathrm{Alg}_1)$ be an ϵ_1-differentially private algorithm and $\mathrm{Alg}_2 : \chi \times \mathrm{range}(\mathrm{Alg}_1) \to \mathrm{range}(\mathrm{Alg}_2)$ be an an ϵ_2-differentially private algorithm. Then $(\mathrm{Alg}_1, \mathrm{Alg}_2 \circ \mathrm{Alg}_1) : \chi \to \mathrm{range}(\mathrm{Alg}_1) \times \mathrm{range}(\mathrm{Alg}_2)$ is $(\epsilon_1 + \epsilon_2)$-differentially private.*

The following Lemma is a variant of *parallel composition* [25] of differential privacy, applied to strings. It says that if we run independent ϵ-differentially private algorithms on disjoint substrings, then the resulting algorithm is still ϵ-differentially private:

▶ **Lemma 12.** *Let Alg_1 and Alg_2 be independent ϵ-differentially private pattern matching algorithms and let S be a string. Further, let $[a, b] \subseteq [0, n-1]$ and $[c, d] \subseteq [0, n-1]$ and $[a, b] \cap [c, d] = \emptyset$. Then algorithm $\mathrm{Alg}_3(S) := (\mathrm{Alg}_1(S[a, b]), \mathrm{Alg}_2(S[c, d]))$ is ϵ-differentially private.*

Proof. Let S and S' be neighbouring strings and let P be a pattern. Let i be the position where $S[i] \neq S'[i]$. Let $\mathrm{Out} = (\mathrm{Out}_1, \mathrm{Out}_2) \subseteq \mathrm{range}(\mathrm{Alg}_1) \times \mathrm{range}(\mathrm{Alg}_2) = \mathrm{range}(\mathrm{Alg}_3)$. If $i \in [a, b]$, then

$$\begin{aligned}
\Pr(\mathrm{Alg}_3(S, P) \in \mathrm{Out}) &= \Pr((\mathrm{Alg}_1(S[a, b], P), \mathrm{Alg}_2(S[c, d], P)) \in (\mathrm{Out}_1, \mathrm{Out}_2)) \\
&= \Pr(\mathrm{Alg}_1(S[a, b], P) \in \mathrm{Out}_1) \cdot \Pr(\mathrm{Alg}_2(S[c, d], P) \in \mathrm{Out}_2) \\
&\leq e^{\epsilon} \cdot \Pr(\mathrm{Alg}_1(S'[a, b], P) \in \mathrm{Out}_1) \cdot \Pr(\mathrm{Alg}_2(S'[c, d], P) \in \mathrm{Out}_2) \\
&= e^{\epsilon} \cdot \Pr(\mathrm{Alg}_3(S', P) \in \mathrm{Out})
\end{aligned}$$

since Alg_1 is ϵ-differentially private and $S[c, d] = S'[c, d]$. The argument for when $i \in [c, d]$ is symmetric. If $i \notin [a, b] \cup [c, d]$, then the output distributions of S and S' are equal. ◀

3 Upper Bounds

In this section we present our differentially private algorithms for the existence, counting and reporting variants of the approximate pattern matching problem.

Algorithm 1 BelowThresh for Approximate Pattern Matching.

```
Input: string $S$, pattern $P$, threshold Thresh, privacy parameter $\epsilon$
Output: a position in $S$ or $\infty$
$m \leftarrow |P|$
$\widetilde{\text{Thresh}} = \text{Thresh} + \text{Lap}(2/\epsilon)$
for $i \in [0, |S| - m]$ do
    $d_i = \text{dist}_H(S[i, i + m - 1], P)$
    $\tilde{d}_i = d_i + \text{Lap}(4/\epsilon)$
    if $\tilde{d}_i \leq \widetilde{\text{Thresh}}$ then
        output $i$
        terminate
    end
end
output $\infty$
```

3.1 The Sparse Vector Technique for Approximate Pattern Matching

Let $\text{Lap}(b)$ denote a random variable drawn from the Laplace distribution with mean 0 and scale b as given in Definition 8. Note that Fact 9 gives a simple algorithm to compute the Hamming distance between $S[i, i + m - 1]$ and P, for any fixed i: Since the sensitivity of $\text{dist}_H(S[i, i + m - 1], P)$ is 1, we can add Laplace noise scaled with $1/\epsilon$, and this gives an additive error of $O(\ln(1/\beta)/\epsilon)$ with probability $1 - \beta$ [15]. However, if we would apply the Laplace mechanism to compute $\text{dist}_H(S[i, i + m - 1], P)$ for *all* $i \in [0, n - m]$, then, since changing one position in S changes up to m of the values of $\text{dist}_H(S[i, i + m - 1], P)$, the sensitivity is m. This results in an additive error of $O((m/\epsilon)\ln(1/\beta))$ with probability $1 - \beta$. Thus, the Laplace mechanism directly applied to this problem is no better than the trivial algorithm of outputting all length-m substrings. Instead, we use a variant of the sparse vector technique (based on an algorithm in [14] and formally described in [15]), which allows to decide for many queries of sensitivity 1 whether the output is above (or in our case, below) a certain threshold, with an error *logarithmic* in the number of queries. Our algorithm for the existence version of the approximate pattern matching problem is given in Algorithm 1. The following two facts follow immediately from [15], chapter 3.6:

▶ **Lemma 13.** *Algorithm 1 is ϵ-differentially private.*

▶ **Lemma 14.** *The output of Algorithm 1 fulfills the following properties with probability $1 - \beta$ and $\alpha = 8\epsilon^{-1}(\ln(|S| - |P| + 1) + \ln(2/\beta))$:*

1. *If Algorithm 1 outputs an index i, then $\text{dist}_H(S[i, i + m - 1], P) \leq \text{Thresh} + \alpha$,*
2. *If i satisfies $\text{dist}_H(S[i, i + m - 1], P) \leq \text{Thresh} - \alpha$ and Algorithm 1 does not terminate before round i, then it outputs i and terminates.*

▶ **Corollary 15.** *There exists an ϵ-dp algorithm solving the existence variant of k-approximate pattern matching with one-sided additive error $\alpha = 16\epsilon^{-1}(\ln(|S| - |P| + 1) + \ln(2/\beta))$ with probability $1 - \beta$.*

Proof. Run Algorithm 1 with $\text{Thresh} = k + 8\epsilon^{-1}(\ln(|S| - |P| + 1) + \ln(2/\beta))$. ◀

3.2 Counting and Reporting

We will distinguish between different cases, depending on whether P is close to a periodic string with a small period or not. We use the following Lemma by Charalampopoulos et al. [8]:

▶ **Lemma 16** (Theorem III.1 in [8]). *Given a pattern P of length m, a string S of length n, and a threshold $k \in [1, \dots, m]$, at least one of the following hold:*

1. *The number of k-mismatch occurrences is bounded by $576 \cdot n/m \cdot k$.*
2. *There exists a (primitive) string Q of length $|Q| \leq \frac{m}{128k}$ that satisfies $\mathrm{dist}_H(Q^\infty[0, m-1], P) \leq 2k$.*

Note that in our privacy definition, only S needs to be private, so we can compute whether case 2 holds for P without losing any privacy. An example of an algorithm computing this is given in Lemma 27 in Appendix A. First, we will consider the case where the pattern P is close to a periodic string with small period, and show that in that case, there is a solution to the reporting problem achieving constant multiplicative error and asymptotically optimal additive error. We will call the different cases the “periodic” and the “non-periodic” case - note that this is not entirely accurate, since the condition says that P is *close* to a periodic string with *small* period. Thus, P can be aperiodic in the periodic case, and P can be periodic, but with a large period, in the non-periodic case.

3.2.1 The Periodic Case

First, we consider the case where a stronger version of condition 2 in Lemma 16 is true for pattern P. In this case we show how to solve the reporting version of the approximate pattern matching problem with constant multiplicative and asymptotically optimal additive error, while satisfying ϵ-differential privacy. We need the following result by Charalampopoulos et al. [8]:

▶ **Lemma 17** (Theorem I.7 in [8]). *Let P denote a pattern of length m, let T denote a text of length $n \leq \frac{3m}{2}$, and let $K \in [0, \dots, m]$ denote a threshold. Suppose that both $T[0, m-1]$ and $T[n-m, n-1]$ are K-mismatch occurrences of P. If there is a positive integer $d \geq 2K$ and a primitive string Q with $|Q| \leq m/(8d)$ and $\mathrm{dist}_H(P, Q^\infty[0, m-1]) \leq d$, then each of the following holds:*

1. *The string T satisfies $\mathrm{dist}_H(T, Q^\infty[0, n-1]) \leq 3d$.*
2. *Every K-mismatch occurrence of P in T starts at a position that is a multiple of $|Q|$.*
3. *The set of all K-mismatch occurrences of P in T can be decomposed into $O(d^2)$ arithmetic progressions with difference $|Q|$.*

The main idea of our algorithm is now the following: first, we divide S into substrings of length at most $\frac{3m}{2}$. Then for each such substring T, we run two instances of Algorithm 1, one for T and P, and one for their reverse strings. If both instances output an occurrence, then with good probability, a substring of T fulfills the conditions of Lemma 17 for a suitable value of $K \geq k$, and we can use the Lemma to report all occurrences of distance at most K. Else, we know by the properties of Algorithm 1 that with good probability, there are no occurrences of distance at most k in T. The details are given in the proof of the following theorem:

Algorithm 2 Reporting Approximate Pattern Matching, periodic case.

```
Input: string $T$, pattern $P$, $|Q|$, $k$, $n$, $m$, $\epsilon$
Output: a set $I$ of positions in $T$
Thresh $= k + \epsilon^{-1}48(\ln(m/2) + \ln(12(n/m)/\beta))$
$\epsilon' = \epsilon/6$
$i \leftarrow$ output of Algorithm 1 on input (string $T$, pattern $P$, threshold Thresh, privacy
  parameter $\epsilon'$)
$j' \leftarrow$ output of Algorithm 1 on input (string $T^{\text{rev}}$, pattern $P^{\text{rev}}$, threshold Thresh,
  privacy parameter $\epsilon'$)
if $j' = \infty$ or $i = \infty$ then
    output $\emptyset$
    terminate
end
$j = (|T| - 1) - j' - (m - 1)$        // translate starting position in $T^{\text{rev}}$ to
  starting position in $T$
output $I = \{i + \ell|Q|, 0 \leq \ell \leq \lfloor \frac{j-i}{|Q|} \rfloor\}$
```

▶ **Theorem 18.** *Let P be a pattern of length m. Assume that there exists a primitive string Q of length $|Q| \leq \frac{m}{32C}$ with $C = \max(k, \frac{96(\ln n + \ln(6/\beta))}{\epsilon})$ that satisfies $\text{dist}_H(P, Q^\infty[0, m-1]) \leq 2k$. Then there exists an $\epsilon-$differentially private algorithm for the reporting version of the k-approximate pattern matching problem, that given a string S of length $n \geq m$ outputs a set $I \subseteq [0, n-m]$ such that with probability $1 - \beta$ the following two conditions are fulfilled:*

1. *If $\text{dist}_H(P, S[i, i+m-1]) \leq k$, then $i \in I$;*
2. *If $i \in I$, then $\text{dist}_H(P, S[i, i+m-1]) \leq (1+\gamma)k + \alpha$,*

where $\gamma = 7$ and $\alpha = 6 \cdot \frac{96(\ln n + \ln(6/\beta))}{\epsilon}$.

Proof. First, we compute a Q satisfying the condition above. Note that we can do unlimited computation on P without violating privacy. An algorithm for computing Q is given in Lemma 27 in Appendix A. Then, we divide the string S into overlapping strings of length at most $\lfloor \frac{3m}{2} \rfloor - 1 = (m-1) + \lfloor \frac{m}{2} \rfloor$. We define $\mathcal{F} = \{S[j \cdot \lfloor m/2 \rfloor, j \cdot \lfloor m/2 \rfloor + \lfloor 3m/2 \rfloor - 2], 0 \leq j \leq \lfloor \frac{n-m}{\lfloor m/2 \rfloor} \rfloor - 1\} \cup [\lfloor \frac{n-m}{\lfloor m/2 \rfloor} \rfloor \lfloor m/2 \rfloor, n-1]$. Note that any two strings in $\mathcal{F}$ overlap by at most $m-1$ and $\mathcal{F}$ covers $[0, n-1]$. Thus, any occurrence of P in S is included in exactly one string $T \in \mathcal{F}$. Further, any position in S is in at most 3 strings in $\mathcal{F}$, and $|\mathcal{F}| \leq n/\lfloor m/2 \rfloor \leq 3n/m$. For every string $T = S[a, b] \in \mathcal{F}$, we run Algorithm 2 and return all positions in $a + I$, where I is the set returned by Algorithm 2 on inputs $(T, P, |Q|, k, n, m, \epsilon)$.

Privacy analysis. Note that in every instance of Algorithm 2, we run two instances of Algorithm 1 with privacy parameter $\epsilon/6$. By Lemma 13 and Fact 11, Algorithm 2 is $\epsilon/3$-differentially private. Further, let S and S' differ in position i^*. Since i^* can only be in at most three strings in $\mathcal{F}$, the full algorithm on S satisfies ϵ-differential privacy by Fact 11 and Lemma 12.

Accuracy analysis. Fix a T in $\mathcal{F}$. Let i and j be as in Algorithm 2 on input T. If j' was set to ∞, let $j = -\infty$. Let $\beta' = \beta/(6(n/m))$. Note that by Lemma 14, with probability at least $1 - \beta'$, we have for all $i' < i$,

$$\begin{aligned}
&\text{dist}_H(T[i', i'+m-1], P) \\
&> k + \epsilon^{-1}48(\ln(m/2) + \ln(12(n/m)/\beta)) - \epsilon^{-1}48(\ln(m/2) + \ln(2/\beta')) \\
&= k, \qquad (2)
\end{aligned}$$

and, if $i < \infty$,

$$\begin{aligned}&\mathrm{dist}_H(T[i, i+m-1], P)\\&\le k + \epsilon^{-1}48(\ln(m/2) + \ln(12(n/m)/\beta)) + \epsilon^{-1}48(\ln(m/2) + \ln(2/\beta'))\\&= k + \epsilon^{-1}96(\ln(6n/\beta)).\end{aligned} \tag{3}$$

Similarly, also with probability $1 - \beta'$, we have for all $i'' > j$,

$$\mathrm{dist}_H(T[i'', i''+m-1], P) > k. \tag{4}$$

and, if $j > -\infty$,

$$\mathrm{dist}_H(T[j, j+m-1], P) \le k + \epsilon^{-1}96(\ln(6n/\beta)). \tag{5}$$

Thus, with probability $1 - \beta/(3(n/m))$, both conditions are true, and since $|\mathcal{F}| \le 3n/m$, these conditions are true with probability at least $1-\beta$ over all instances of Algorithm 2. In the following, we condition on that.

If either j' or i was set to ∞, then there is no occurrence of distance at most k in T, and in this case we return the empty set. Next, if $j < i$, then there is also no occurrence of at most k in T by (2) and (4). Note that also in this case, Algorithm 2 returns the empty set.

Now, consider the case $j \ge i$ for finite integers i and j. We want to argue that in this case, the string $T[i, j+m-1]$ fulfills the conditions of Lemma 17 for an appropriate choice of $K > k$. Obviously, $|T[i, j+m-1]| \le |T| \le \frac{3m}{2}$. We set $K = k + \epsilon^{-1}96(\ln(6n/\beta)) \le 2C$. By (3) and (5) both i and j are the start of a K-mismatch occurrence. Let $d = 2K \le 4C$. By assumption, there is a primitive string Q with $|Q| \le m/(32C) \le m/8d$ with $\mathrm{dist}_H(P, Q^\infty[0, m-1]) \le 2k \le d$. Thus, the conditions of Lemma 17 are fulfilled. This gives the following:

1. Since the string $T[i, j+m-1]$ satisfies $\mathrm{dist}_H(T[i, j+m-1], Q^\infty[0, j+m-i-1]) \le 3d$, we have that for any position $q = i + \ell|Q|$ for $\ell \in [0, \lfloor\frac{j-i}{|Q|}\rfloor]$:

$$\begin{aligned}&\mathrm{dist}_H(T[q, q+m-1], P)\\&\le \mathrm{dist}_H(T[q, q+m-1], Q^\infty[0, m-1]) + \mathrm{dist}_H(Q^\infty[0, m-1], P)\\&\le 3d + 2k = 8k + 6 \cdot 96 \cdot \epsilon^{-1}(\ln(6n/\beta)).\end{aligned}$$

   Thus, every reported occurrence q fulfills $\mathrm{dist}_H(T[q, q+m-1], P) \le (1+\gamma)k + \alpha$ with $\gamma = 7$ and $\alpha = 6 \cdot 96 \cdot \epsilon^{-1}(\ln(6n/\beta))$.

2. Since every K-mismatch occurrence of P in $T[i, j+m-1]$ starts at a multiple of $|Q|$, then in particular, any k-mismatch occurrence of P in $T[i, j+m-1]$ starts at a position $i + \ell|Q|$ in T for $\ell \in [0, \lfloor\frac{j-i}{|Q|}\rfloor]$. Thus, any substring of $T[i, j+m-1]$ of length m that does not start at $i + \ell|Q|$ for some $\ell \in [0, \lfloor\frac{j-i}{|Q|}\rfloor]$ has a distance larger than k.

Further, by (2) and (4), $\mathrm{dist}_H(T[i', i'+m-1], P) > k$ for all $i' < i$ or $i' > j$. Thus, we report all occurrences with distance at most k. ◀

3.2.2 The Non-Periodic Case

Next, we assume condition 2 in Lemma 16 is not true for P, that is, there does not exist a string Q of length $|Q| \le \frac{m}{128k}$ that satisfies $\mathrm{dist}_H(Q^\infty[0, m-1], P) \le 2k$. This means the number of k-mismatch occurrences in any string T of length $|T|$ is bounded by $576 \cdot |T|/m \cdot k$ by Lemma 16. In particular, in any substring of length $\le 2m$ of S, the number of occurrences is at most $1152k = O(k)$. We will use this fact to solve the counting variant of the problem in the non-periodic case. Note that Theorem 18 and Theorem 19 do not cover all the cases: If $k \le C/4$, where C is as in Theorem 18, then it is possible that the conditions of neither theorem are fulfilled. We deal with that case later.

Algorithm 3 Counting Approximate Pattern Matching, non-periodic case.

```
Input: string $T$, pattern $P$, $k$, $n$, $m$, $\epsilon$
Output: a count $c$ and a position $j$ in $T$
$j = -1$
$i = -1$
$c = 0$
Thresh $= k + \epsilon^{-1}16 \cdot 1152k(\ln m + \ln(2(n/m)1152k/\beta))$
$\epsilon' = \epsilon/(2 \cdot 1152k)$
while $i < |T| - |P|$ & $c < 1152k$ do
    $j \leftarrow$ output of Algorithm 1 on input (string $T[i+1, n-1]$, pattern $P$, threshold
     Thresh, privacy parameter $\epsilon'$)
    if $j = \infty$ then
        output $(c, i)$
        terminate
    end
    $c = c + 1$
    $i = j$
end
output $(c, j)$
```

▶ **Theorem 19.** *Let P be a pattern of length m. If there does not exist a string Q of length $|Q| \leq \frac{m}{128k}$ that satisfies $\mathrm{dist}_H(Q^\infty[0, m-1], P) \leq 2k$, then there exists an $\epsilon-$differentially private algorithm that given a string S of length $n \geq m$ computes a count c, such that with probability $1 - \beta$ it holds that $c_k(S) \leq c \leq c_{(1+\gamma)k}(S)$, where $\gamma = O(\epsilon^{-1} \cdot (\ln n + \ln(1/\beta))$. Further, if $c > 0$, it returns a witness i satisfying $\mathrm{dist}_H(P, S[i, i+m-1]) \leq (1+\gamma)k$.*

Proof. The first step is to divide the string S into substrings of length at most $2m-1$, which form overlapping blocks, such that any pattern occurrence appears in exactly one block. That is, we define the set $\mathcal{B} = \{S[jm, (j+2)m-2], j = 0 \ldots \lfloor \frac{n+1}{m} \rfloor - 2\} \cup \{S[(\lfloor \frac{n+1}{m} \rfloor - 1)m, n-1]\}$. Since $\mathcal{B}$ covers $[0, n-1]$ and two strings overlap by at most $m-1$, any pattern occurrence in S is contained in exactly one string in $\mathcal{B}$. Note that any position in S is included in at most two strings in $\mathcal{B}$.

For each $T \in \mathcal{B}$, we run Algorithm 3. Then for the outputs $(c_1, j_1), \ldots, (c_{|\mathcal{B}|}, j_{|\mathcal{B}|})$, we output $\sum_{\ell=1}^{|\mathcal{B}|} c_\ell$. If there exists a $j_\ell > -1$, we choose an arbitrary such and output $\ell m + j_\ell$.

Privacy analysis. For any instance of Algorithm 3, we run at most $1152k$ instances of Algorithm 1 with privacy parameter $\epsilon' = \epsilon/(2 \cdot 1152k)$. Thus any instance of Algorithm 3 is $\epsilon/2$-differentially private by Lemma 13 and Fact 11. Further, let S and S' differ in position i^*. Since i^* can only be in at most two strings in $\mathcal{B}$, the full algorithm satisfies ϵ-differential privacy by Fact 11 and Lemma 12.

Accuracy analysis. Let $c(T)$ be the output of Algorithm 3 for string $T \in \mathcal{B}$ and $c_k(T)$ the true count of positions i such that $\mathrm{dist}_H(T[i, i+m-1], P) \leq k$. For a fixed T, we will show that $c_k(T) \leq c(T) \leq c_{(1+\gamma)k}(T)$ with probability $1 - \beta/(n/m)$. Since $|\mathcal{B}| \leq n/m$, a union bound then implies that the bound holds for all $T \in \mathcal{B}$ with probability $1 - \beta$. Note that since any substring of length m of S is included in exactly one string in $\mathcal{B}$, this implies $c_k(S) = \sum_{T \in \mathcal{B}} c_k(T) \leq \sum_{T \in \mathcal{B}} c(T) \leq \sum_{T \in \mathcal{B}} c_{(1+\gamma)k}(T) = c_{(1+\gamma)k}(S)$.

Now, fix $T \in \mathcal{B}$ and let $\alpha' = 8(\epsilon')^{-1}(\ln(|T| - |P| + 1) + \ln(2/\beta'))$. By Lemma 14, with probability at least $1 - \beta'$, whenever an instance of Algorithm 1 in Algorithm 3 returns a position i, the distance $\mathrm{dist}_H(T[i, i+m-1], P) \leq \mathrm{Thresh} + \alpha'$; further, any position $i' \leq i$

which was part of that instance satisfies $\mathrm{dist}_H(T[i', i'+m-1], P) > \mathrm{Thresh} - \alpha'$ (otherwise it would have been output instead of i). Thus, for each such i and $\beta' = \beta/((n/m)1152k)$ we have

$$\begin{aligned}\mathrm{dist}_H(T[i, i+m-1], P) &\leq k{+}\epsilon^{-1}16 \cdot 1152k(\ln m + \ln(2(n/m)1152k/\beta)) + \alpha' \\ &= k{+}\epsilon^{-1}16 \cdot 1152k(\ln m + \ln(2(n/m)1152k/\beta)) \\ &\quad +8(\epsilon')^{-1}(\ln(|T| - |P| + 1) + \ln(2/\beta')) \\ &= k{+}\epsilon^{-1}16 \cdot 1152k(\ln m + \ln(2(n/m)1152k/\beta)) \\ &\quad\ +\epsilon^{-1}16 \cdot 1152k(\ln m + \ln(2/\beta')) \\ &= k{+}2\alpha',\end{aligned}$$

and for each $i' \leq i$ in that instance of Algorithm 1

$$\begin{aligned}\mathrm{dist}_H(T[i', i'+m-1], P) &> k + \epsilon^{-1}16 \cdot 1152k(\ln m + \ln(2(n/m)1152k/\beta)) - \alpha' \\ &= k,\end{aligned}$$

with probability $1 - \beta'$. Thus, over the entire run of Algorithm 3, the inequalities hold with probability at least $1 - \beta/(n/m)$, and we condition on that. It directly follows that all counted positions i satisfy $\mathrm{dist}_H(T[i, i+m-1], P) \leq k + 2\alpha' = (1+\gamma)k$, for $\gamma = \epsilon^{-1}32 \cdot 1152(\ln m + \ln(2(n/m)1152k/\beta) = O(\epsilon^{-1}(\ln n + \ln(1/\beta)))$. Thus, $c(T) \leq c_{(1+\gamma)k}(T)$. For the lower bound, there are two cases to consider:

Case 1: If $c < 1152k$ when Algorithm 3 ends, then every possible starting position $i \leq |T| - |P|$ was considered by some instance of Algorithm 1. Thus, all positions i satisfying $\mathrm{dist}_H(T[i, i+m-1], P) \leq k$ were counted and $c_k(T) \leq c(T) \leq c_{(1+\gamma)k}(T)$.

Case 2: If $c = 1152k$, then $c_k(T) \leq 1152k$ holds by Lemma 16 and since $|T| < 2m$. ◀

3.2.3 Non-periodic and small k

Note that there can be a case where neither the conditions of Theorem 19 nor Theorem 18 are fulfilled: If $k < C/4 = 24\epsilon^{-1}\ln(6n/\beta)$, and there exists a primitive string Q of length $|Q| \leq m/(128k)$ such that $\mathrm{dist}_H(P, Q^\infty[0, m-1]) \leq 2k$, but there does not exist a primitive string Q' of length $|Q'| \leq m/(32C)$ such that $\mathrm{dist}_H(P, Q'^\infty[0, m-1]) \leq 2k$. Note that the second condition implies that there does not exist a primitive string Q' of length $|Q'| \leq m/(128K)$ such that $\mathrm{dist}_H(P, Q'^\infty[0, m-1]) \leq 2k < 2K$, for $K = C/4$.

▶ **Lemma 20.** *Let P be a pattern of length m. If $k < K = 24\epsilon^{-1}\ln(6n/\beta)$ and there does not exist a string Q of length $|Q| \leq m/(128K)$ such that $\mathrm{dist}_H(P, Q^\infty[0, m-1]) \leq 2K$, then there exists an ϵ-differentially private algorithm that given a string S of length $n \geq m$ computes a count c, such that with probability $1 - \beta$ it holds that $c_k(S) \leq c \leq c_{k+\alpha}(S)$, where $\alpha = O(\epsilon^{-2}(\ln^2(n/\beta)))$.*

Proof. Note that the conditions of Theorem 19 are fulfilled with K taking the role of k. Thus there exists an algorithm that outputs a count c such that with probability $1-\beta$ it holds that $c_K(S) \leq c \leq c_{(1+\gamma)K}(S)$ where $\gamma = O(\epsilon^{-1}(\ln(n/\beta)))$. The lemma now follows since $c_k(S) \leq c_K(S)$ and $c_{(1+\gamma)K}(S) \leq c_{\eta^2}(S) \leq c_{k+\eta^2}(S)$ for $\eta = \max(1+\gamma, K) = O(\epsilon^{-1}(\ln(n/\beta)))$. ◀

Theorem 1 now follows by noticing that any pattern P fulfills the conditions of either Theorem 18, Theorem 19 or Lemma 20, and that the reporting solution from Theorem 18 implies a counting solution with the same error bounds.

4 Lower Bound

For any $k \leq m$, there is a trivial algorithm solving the reporting version of the approximate pattern matching problem with additive one-sided error $O(m-k)$ with probability 1 while preserving ϵ-differential privacy: We just output every position $i \in [0, n-m+1]$. The next Theorem shows that in order to have error $o(m-k)$, we need $m = \Omega(\ln n)$, and in that case the additive error is $\Omega(\ln(n/m))$. Note that the lower bound holds for any pattern P and for the existence or counting variant, as long as at least one witness is returned. Our lower bound is based on a packing argument.

▶ **Theorem 21.** *Let P be any string of length m and let $k < m$ be a parameter. Assume there is an ϵ-differentially private algorithm* Alg *with the following guarantee: If S is a string of length $n \geq m$ such that there exists $j \in [0, n-m]$ with $\mathrm{dist}_H(S[j, j+m-1], P) \leq k$, then with probability at least $2/3$,* Alg *returns a position $i \in [0, n-m]$ such that $\mathrm{dist}_H(S[i, i+m-1], P) \leq k+\alpha$. Then either $\alpha = \Omega(m-k)$, or $m = \Omega(\epsilon^{-1}\ln n)$ and $\alpha = \Omega(\epsilon^{-1}\ln(n/m))$.*

Proof. First, we assume there is an algorithm Alg as in the statement of the theorem satisfying $\alpha < m-k$. We show $m = \Omega(\ln n)$. We start by dividing $[0, n-1]$ into disjoint intervals of length m (we assume wlog that n is a multiple of m). That is, we define the set $\mathcal{I} = \{[jm, (j+1)m-1], j = 0, \ldots, n/m-1\}$. For every *even* $j \in \{0, \ldots, n/m-1\}$, we define a string S_j as follows: $S_j[jm, (j+1)m-1] = P$, and for all $q \in [0, n-1] \backslash [jm, (j+1)m-1]$, we set $S_j[q] = \$$ for some \$ which does not appear in P.

Note that S_j and S_i have a Hamming distance of $2m$ for all even $i \neq j$, $i, j \in [0, n/m-1]$. Further, we have $\mathrm{dist}_H(S_j[jm, (j+1)m-1], P) = 0 \leq k$, and for every $q \in [0, n-m] \backslash [(j-1)m+1, (j+1)m-1]$, we have $\mathrm{dist}_H(S_j[q, q+m-1], P) = m > k+\alpha$. Thus, by assumption on Alg, we have

$$\Pr(\mathrm{Alg}(S_j) \in [(j-1)m+1, (j+1)m-1]) \geq 2/3,$$

and, by group privacy (Fact 10),

$$\Pr(\mathrm{Alg}(S_j) \in [(i-1)m+1, (i+1)m-1]) \geq e^{-2m\epsilon} 2/3,$$

for every even $i \in [0, n/m-1]$. Since these events are disjoint, we have

$$1 \geq \sum_{\text{even } i \in [0, n/m-1]} e^{-2m\epsilon} 2/3$$

and therefore

$$m \geq (2\epsilon)^{-1}(\ln(n/(2m)) + \ln(2/3)),$$

and therefore $m = \Omega(\epsilon^{-1}\ln n)$.

Next, we want to show $\alpha = \Omega(\ln(n/m))$. For this, we consider the same partition $\mathcal{I}$ into intervals, and for every even j in $[0, n/m-1]$ we define S_j as follows: $S_j[jm, (j+1)m-1] = \$^k P[k, m-1]$, and for every even $i \neq j$, $i \in [0, n/m-1]$, we define $S_j[im, (i+1)m-1] = \$^{k+\alpha+1} P[k+\alpha+1, m-1]$. For all other positions $q \in [0, n-1]$, we define $S_j[q] = \$$. We have $\mathrm{dist}_H(S_j[jm, (j+1)m-1], P) \leq k$ and $\mathrm{dist}_H(S_j[q, q+m-1], P) > k+\alpha$ for all $q \in [0, n-m] \backslash [(j-1)m+1, (j+1)m-1]$. Further, all S_j, S_i with i, j even and $i \neq j$ have a Hamming distance of $2\alpha+2$. By assumption on Alg we have

$$\Pr(\mathrm{Alg}(S_j) \in [(j-1)m+1, (j+1)m-1]) \geq 2/3,$$

and, by group privacy (Fact 10),

$$\Pr(\mathrm{Alg}(S_j) \in [(i-1)m+1, (i+1)m-1]) \geq e^{-(2\alpha+2)\epsilon}2/3.$$

for every even $i \in [0, n/m-1]$. Since these events are disjoint, we have

$$1 \geq \sum_{\text{even } i \in [0,n/m-1]} e^{-(2\alpha+2)\epsilon}2/3$$

and therefore

$$\alpha \geq (2\epsilon)^{-1}(\ln(n/2m) + \ln(2/3)) - 1,$$

and therefore $\alpha = \Omega(\epsilon^{-1}\ln(n/m))$. ◀

5 Conclusion

We have initiated a study of *differentially private pattern matching* algorithms, and have shown that combining techniques from the areas of differential privacy and pattern matching can be used to obtain interesting new results. Specifically, for the approximate pattern matching problem with Hamming distance under ϵ-differential privacy, we have both shown a strong lower bound and new upper bounds. The upper bounds asymptotically match the lower bound for the existence variant, and for the reporting variant for a special class of patterns. There are many potential directions for future research, including:

- closing the gap between the upper and the lower bound for *all* patterns;
- studying (ϵ, δ)-differential privacy for this problem;
- considering other distance measures, e.g. edit distance, both for the definition of k-approximate pattern matching, and for the privacy definition;
- considering other error measures, e.g. for the counting variant of pattern matching.

Further, it would be exciting to see if it is possible to obtain differentially private indexing data structures with useful error guarantees.

References

1 Oluwole I. Ajala, Hayam Alamro, Costas S. Iliopoulos, and Grigorios Loukides. Towards string sanitization. In *Proc. 14th AIAI (Workshops)*, pages 200–210, 2018. doi:10.1007/978-3-319-92016-0_19.

2 Giulia Bernardini, Huiping Chen, Alessio Conte, Roberto Grossi, Grigorios Loukides, Nadia Pisanti, Solon P. Pissis, Giovanna Rosone, and Michelle Sweering. Combinatorial algorithms for string sanitization. *ACM Trans. Knowl. Discov. Data*, 15(1):8:1–8:34, 2021. doi:10.1145/3418683.

3 Giulia Bernardini, Alessio Conte, Garance Gourdel, Roberto Grossi, Grigorios Loukides, Nadia Pisanti, Solon P. Pissis, Giulia Punzi, Leen Stougie, and Michelle Sweering. Hide and mine in strings: Hardness, algorithms, and experiments. *IEEE Trans. Knowl. Data Eng.*, 35(6):5948–5963, 2023. doi:10.1109/TKDE.2022.3158063.

4 Giulia Bernardini, Nadia Pisanti, Solon P. Pissis, and Giovanna Rosone. Approximate pattern matching on elastic-degenerate text. *Theor. Comput. Sci.*, 812:109–122, 2020. doi:10.1016/j.tcs.2019.08.012.

5 Raghav Bhaskar, Srivatsan Laxman, Adam D. Smith, and Abhradeep Thakurta. Discovering frequent patterns in sensitive data. In *Proc. 16th SIGKDD*, pages 503–512, 2010. doi:10.1145/1835804.1835869.

6 Luca Bonomi and Li Xiong. A two-phase algorithm for mining sequential patterns with differential privacy. In *Proc. 22nd CIKM*, pages 269–278, 2013. doi:10.1145/2505515.2505553.

7 Luca Bonomi, Li Xiong, Rui Chen, and Benjamin C. M. Fung. Frequent grams based embedding for privacy preserving record linkage. In *Proc. 21st CIKM*, pages 1597–1601, 2012. doi:10.1145/2396761.2398480.

8 Panagiotis Charalampopoulos, Tomasz Kociumaka, and Philip Wellnitz. Faster approximate pattern matching: A unified approach. In *Proc. 61st FOCS*, pages 978–989, 2020.

9 Rui Chen, Gergely Ács, and Claude Castelluccia. Differentially private sequential data publication via variable-length n-grams. In *Proc. 19th CCS*, pages 638–649, 2012. doi:10.1145/2382196.2382263.

10 Rui Chen, Benjamin C. M. Fung, Bipin C. Desai, and Nériah M. Sossou. Differentially private transit data publication: a case study on the montreal transportation system. In *Proc. 18th KDD*, pages 213–221, 2012. doi:10.1145/2339530.2339564.

11 Rui Chen, Benjamin C. M. Fung, Noman Mohammed, Bipin C. Desai, and Ke Wang. Privacy-preserving trajectory data publishing by local suppression. *Inf. Sci.*, 231:83–97, 2013. doi:10.1016/j.ins.2011.07.035.

12 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In *Proc. 3rd TCC*, volume 3876, pages 265–284, 2006. doi:10.1007/11681878_14.

13 Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In Leonard J. Schulman, editor, *Proc. 42nd STOC*, pages 715–724, 2010.

14 Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil P. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proc. 41st STOC*, pages 381–390, 2009. doi:10.1145/1536414.1536467.

15 Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

16 Hendrik Fichtenberger, Monika Henzinger, and Jalaj Upadhyay. Constant matters: Fine-grained error bound on differentially private continual observation. In *Proc. 40th ICML*, 2023.

17 Pawel Gawrychowski and Przemyslaw Uznanski. Towards unified approximate pattern matching for hamming and l_1 distance. In *Proc. 45th ICALP*, pages 62:1–62:13, 2018. doi:10.4230/LIPIcs.ICALP.2018.62.

18 Xi He, Graham Cormode, Ashwin Machanavajjhala, Cecilia M. Procopiuc, and Divesh Srivastava. DPT: differentially private trajectory synthesis using hierarchical reference systems. *Proc. VLDB Endow.*, 8(11):1154–1165, 2015. URL: http://www.vldb.org/pvldb/vol8/p1154-he.pdf.

19 Kunho Kim, Sivakanth Gopi, Janardhan Kulkarni, and Sergey Yekhanin. Differentially private n-gram extraction. In *Proc. 34th NeurIPS*, pages 5102–5111, 2021. URL: https://proceedings.neurips.cc/paper/2021/hash/28ce9bc954876829eeb56ff46da8e1ab-Abstract.html.

20 Elahe Ghasemi Komishani, Mahdi Abadi, and Fatemeh Deldar. PPTD: preserving personalized privacy in trajectory data publishing by sensitive attribute generalization and trajectory local suppression. *Knowl. Based Syst.*, 94:43–59, 2016. doi:10.1016/j.knosys.2015.11.007.

21 Yanhui Li, Guoren Wang, Ye Yuan, Xin Cao, Long Yuan, and Xuemin Lin. Privts: Differentially private frequent time-constrained sequential pattern mining. In *Proc. 23rd DASFAA*, pages 92–111, 2018. doi:10.1007/978-3-319-91458-9_6.

22 Md Safiur Rahman Mahdi, Md Momin Al Aziz, Noman Mohammed, and Xiaoqian Jiang. Privacy-preserving string search on encrypted genomic data using a generalized suffix tree. *Informatics in Medicine Unlocked*, 23:100525, 2021.

23 Nicholas Mainardi, Alessandro Barenghi, and Gerardo Pelosi. Privacy preserving substring search protocol with polylogarithmic communication cost. In *Proc. 35th ACSAC*, pages 297–312, 2019. doi:10.1145/3359789.3359842.

24 Mihai Maruseac and Gabriel Ghinita. Differentially-private mining of representative travel patterns. In *Proc. 17th MDM*, pages 272–281, 2016. doi:10.1109/MDM.2016.48.

25 Frank McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. *Commun. ACM*, 53(9):89–97, 2010. doi:10.1145/1810891.1810916.

26 Gonzalo Navarro. A guided tour to approximate string matching. *ACM Comput. Surv.*, 33(1):31–88, 2001. doi:10.1145/375360.375365.

27 Shiyue Qin, Fucai Zhou, Zongye Zhang, and Zifeng Xu. Privacy-preserving substring search on multi-source encrypted gene data. *IEEE Access*, 8:50472–50484, 2020. doi:10.1109/ACCESS.2020.2980375.

28 Kana Shimizu, Koji Nuida, and Gunnar Rätsch. Efficient privacy-preserving string search and an application in genomics. *Bioinform.*, 32(11):1652–1661, 2016. doi:10.1093/bioinformatics/btw050.

29 Tatiana Starikovskaya. Communication and streaming complexity of approximate pattern matching. In Juha Kärkkäinen, Jakub Radoszewski, and Wojciech Rytter, editors, *Proc. 28th CPM*, pages 13:1–13:11, 2017. doi:10.4230/LIPIcs.CPM.2017.13.

30 Hiroki Sudo, Masanobu Jimbo, Koji Nuida, and Kana Shimizu. Secure wavelet matrix: Alphabet-friendly privacy-preserving string search for bioinformatics. *IEEE ACM Trans. Comput. Biol. Bioinform.*, 16(5):1675–1684, 2019. doi:10.1109/TCBB.2018.2814039.

31 Juan Ramón Troncoso-Pastoriza, Stefan Katzenbeisser, and Mehmet Utku Celik. Privacy preserving error resilient dna searching through oblivious automata. In *Proc. 14th CCS*, pages 519–528, 2007. doi:10.1145/1315245.1315309.

32 Sirintra Vaiwsri, Thilina Ranbaduge, and Peter Christen. Accurate and efficient privacy-preserving string matching. *Int. J. Data Sci. Anal.*, 14(2):191–215, 2022. doi:10.1007/s41060-022-00320-5.

33 Zhibo Wang, Wenxin Liu, Xiaoyi Pang, Ju Ren, Zhe Liu, and Yongle Chen. Towards pattern-aware privacy-preserving real-time data collection. In *Proc. 39th INFOCOM*, pages 109–118, 2020. doi:10.1109/INFOCOM41043.2020.9155290.

34 Jianhao Wei, Yaping Lin, Xin Yao, Jin Zhang, and Xinbo Liu. Differential privacy-based genetic matching in personalized medicine. *IEEE Trans. Emerg. Top. Comput.*, 9(3):1109–1125, 2021. doi:10.1109/TETC.2020.2970094.

35 Xiaochao Wei, Minghao Zhao, and Qiuliang Xu. Efficient and secure outsourced approximate pattern matching protocol. *Soft Comput.*, 22(4):1175–1187, 2018. doi:10.1007/s00500-017-2560-4.

36 Xinyu Yang, Teng Wang, Xuebin Ren, and Wei Yu. Survey on improving data utility in differentially private sequential data publishing. *IEEE Trans. Big Data*, 7(4):729–749, 2021. doi:10.1109/TBDATA.2017.2715334.

37 Jun Zhang, Xiaokui Xiao, and Xing Xie. Privtree: A differentially private algorithm for hierarchical decompositions. In *Proc. ACM SIGMOD*, pages 155–170, 2016. doi:10.1145/2882903.2882928.

38 Peng Zhang and Mikhail J. Atallah. On approximate pattern matching with thresholds. *Inf. Process. Lett.*, 123:21–26, 2017. doi:10.1016/j.ipl.2017.03.001.

A Runtime Analysis

In the following, we analyze the runtime of our algorithms and show that it is $O(nm + m^3)$, assuming that noises from the Laplace distribution can be drawn in constant time. We note that in this work we did not optimize for runtime.

First, note that computing the Hamming distance between $S[i, i + m - 1]$ and P for any i can be done in m time. We collect some immediate observations about the runtimes of the given algorithms, if we already know whether P fulfills the conditions of the theorems (and for which $|Q|$).

▶ **Lemma 22.** *Let i be the output of Algorithm 1 on an input string T and pattern P. The runtime of Algorithm 1 is $O(\min(i \cdot m, |T| \cdot m))$.*

▶ **Corollary 23.** *The runtime of Algorithm 2 on input string T and pattern P is $O(|T| \cdot m)$.*

▶ **Corollary 24.** *The runtime of Algorithm 3 on input string T and pattern P is $O(|T| \cdot m)$.*

▶ **Corollary 25.** *Given P and $|Q|$ satisfying the conditions of Theorem 18, the algorithm given by Theorem 18 has a runtime of $O(nm)$.*

▶ **Corollary 26.** *The algorithms of Theorem 19 and Lemma 20 have a runtime of $O(nm)$.*

Next, we analyze the "preprocessing" part for P, i.e. we show how to decide if P is close to a periodic string Q^∞ with small $|Q|$.

▶ **Lemma 27.** *Let P be a pattern of length m and let k be a parameter. In $O(m^3)$ time, we can decide if there exists a Q such that $|Q| \leq \max(\frac{m}{32C}, \frac{m}{128k})$ fulfilling $\mathrm{dist}_H(P, Q^\infty[0, m-1]) \leq 2k$, where C is defined as in Theorem 18, and compute the shortest such.*

Proof. For any potential $q \leq \max(\frac{m}{32C}, \frac{m}{128k})$, we do the following: First, we conceptually partition the pattern P into blocks of length q. Note that there are at least $m/q \geq \min(128k, 32C) \geq 32k$ such blocks. Now assume there exists Q of length $|Q| = q$ satisfying $\mathrm{dist}_H(P, Q^\infty[0, m-1]) \leq 2k$. Then, since $\mathrm{dist}_H(P, Q^\infty[0, m-1]) \leq 2k$, *all but at most* $2k$ *blocks* of P have to be equal to Q. Note that there can be at most one potential string of length q fulfilling that condition. To find it, we traverse P and count how often a block in P is equal to any given substring of length q. We can do this by e.g. building a trie of all blocks as we traverse P. This takes $O(m)$ time. Now, if we found a candidate string Q such that all but at most $2k$ blocks are equal to Q, we spend at most m time to check if indeed $\mathrm{dist}_H(P, Q^\infty[0, m-1]) \leq 2k$. Since there are at most $\max(\frac{m}{32C}, \frac{m}{128k}) \leq m$ possible values of q, the total runtime is $O(m^3)$. ◀

Note that the condition of Lemma 20 can be checked by applying Lemma 27 with $C/4$ taking the role of k.