Sequence Obfuscation to Thwart Pattern Matching Attacks

Bo Guan*, Nazanin Takbiri*, Dennis L. Goeckel*, Amir Houmansadr†, Hossein Pishro-Nik*,
*Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA, 01003 USA
{boguan, ntakbiri, goeckel, pishro}@ecs.umass.edu
†College of Information and Computer Sciences, University of Massachusetts, Amherst, MA, 01003 USA
amir@cs.umass.edu

Abstract—**Suppose we are given a large number of sequences on a given alphabet, and an adversary is interested in identifying (de-anonymizing) a specific target sequence based on its patterns. Our goal is to thwart such an adversary by obfuscating the target sequences by applying artificial (but small) distortions to its values. A key point here is that we would like to make no assumptions about the statistical model of such sequences. This is in contrast to existing literature where assumptions (e.g., Markov chains) are made regarding such sequences to obtain privacy guarantees. We relate this problem to a set of combinatorial questions on sequence construction based on which we are able to obtain provable guarantees. This problem is relevant to important privacy applications: from fingerprinting webpages visited by users through anonymous communication systems to linking communicating parties on messaging applications to inferring activities of users of IoT devices.**

Index Terms—**Anonymization, information-theoretic privacy, Internet of Things (IoT), obfuscation, Privacy Preserving Mechanism (PPM), statistical matching, superstring.**

I. Introduction

Consider a scenario where n length-m sequences are being generated. You do not know anything about the way the sequences are constructed (e.g., the distribution of the data points in the sequences). All you know is that these n sequences, labeled $1, 2, \dots, n$, each have m elements drawn from the same size-r alphabet-$\mathcal{R}$. The sequences are revealed to a potential adversary but the labels are hidden. However, the adversary has obtained a “pattern" (later defined precisely) in one of these sequences and would like to identify that sequence based on that pattern. Your job is to design an obfuscation mechanism to apply to these sequences before they are revealed to the adversary to prevent such identification. The difficulty is that you have very limited knowledge about these sequences (the values of n, m, and $\mathcal{R}$), and you do not know which sequence the adversary might be seeking to identify or what pattern the adversary might employ. And, of course, we want the distortion to be as small as possible.

The considered problem addresses key scenarios in privacy or security: fingerprinting webpages visited by users through anonymous communication systems, linking communicating parties on messaging applications, and inferring the activities of the users of IoT devices. While the setting is general, to have a concrete example in mind, we motivate the problem from the consideration of *User-Data Driven* (UDD) services in IoT applications: data submitted by users is analyzed with the goal of improving the service in applications such as health care, smart homes, and connected vehicles. This tuning by UDD services has significantly improved customers' lives, but the large amount of user data collected by these applications can compromise users' privacy. These privacy and security threats are a major obstacle to the adoption of IoT applications [1]–[11]. In order to improve users' privacy, anonymization and obfuscation mechanisms are used at the cost of user utility. The anonymization technique frequently changes the pseudonym of user mappings in order to reduce the length of time series that can be utilized for statistical analysis [12]–[18]. By contrast, obfuscation adds noise to users' data samples to increase users' privacy [19]–[25].

To provide a privacy guarantee in sequence matching analyses, a general stochastic model for the users' data (e.g., Markov chains) has been generally assumed [26]–[31]. However, as *Privacy-Protection Mechanism* (PPM) designers, we may not know the underlying statistical model for users' data. Takbiri et al. [24] have shown that modeling errors can destroy privacy guarantees: a privacy mechanism that provides perfect privacy under one statistical model may break down under another model, even if a very high level of obfuscation is employed. Hence, an important question is whether we can provide robust privacy mechanisms without assuming a certain model for users' data.

In practice, many privacy attacks are based on simple “pattern matching" [32], [33], where the adversary looks for an ordered sequence of values that appear close to each other in the user's data. Our goal is to provide privacy guarantees, even if we do not know what patterns the adversaries might be exploiting. By focusing on this common type of privacy attack (pattern matching), we are able to eliminate the need for making specific assumptions about the users' data model. In other words, we are able to seek a model-free approach by focusing on a specific (albeit very common) type of attack.

Our obfuscation approach is based on the following idea: noise should be added in a way that the obfuscated data sequences are likely to have a large number of common patterns. This means that for any user and for any potential pattern that the adversary might obtain for that user, there will

This work was supported by the National Science Foundation under grants CCF–1421957 and CNS–1739462.

be a large number of other users with the same data pattern in their obfuscated sequences. This in turn can be used to provide privacy guarantees against pattern matching attacks.

To achieve privacy guarantees we rely on the concept of *superstrings*, which contain every possible pattern of length less than or equal to l. This in turn happens to be related to a rich area in combinatorics [34]–[37]. Of relevant interest are the De Bruijn sequences [38], which give the answer for the shortest cyclic sequence in which repeated symbols are allowed in each contiguous substring when the substring length is restricted to be less than the size of the alphabet.

The setting of our privacy guarantees can be summarized as follows: (i) we make no assumption on the statistical model of users' sequence except for the alphabet-$\mathcal{R}$; (ii) we make no assumption about the pattern known to the adversary except for its length-l; (iii) we obtain non-asymptotic guarantees.

II. System Model, Definitions, and Metrics

Here, we employ a framework similar to [18], [24], [39], [40]. The system has n users with $X_u(k)$ denoting the data of user u at time k, which we would like to protect from an interested adversary. We also assume there are $r \geq 2$ possible values for each user's data points in a finite size set $\mathcal{R} = \{0, 1, \ldots, r-1\}$.

As shown in Fig. 1, in order to achieve privacy for users, both anonymization and obfuscation techniques are employed. In Fig. 1, $Z_u(k)$ denotes the reported data point of user u at time k after applying the obfuscation, and $Y_u(k)$ denotes the reported data point of user u at time k after applying the obfuscation and the anonymization.

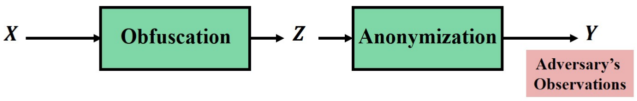

Fig. 1: Applying obfuscation and anonymization techniques to the users' data points.

Data Points Model: Let $\mathbf{X}_u$ be the $m \times 1$ vector containing the data points of user u, and $\mathbf{X}$ be the $m \times n$ matrix with the u^{th} column equal to $\mathbf{X}_u$:

$$\mathbf{X}_u = [X_u(1), X_u(2), \cdots, X_u(m)]^T, \quad \mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_n].$$

Next, we provide a formal definition of a *pattern* to provide a suitable model for a typical pattern matching attack. For instance, a potential pattern could be the sequence of locations that the user normally visits in a particular order. The visited locations might not necessarily be contiguous in the sequence, but they are close to each other in time. Hence, we impose two conditions on a *pattern*: first, the elements of the pattern sequence must be present in order. Second, consecutive elements of the pattern sequence must appear close to each other, with *distance* less than or equal to h, where the *distance* between two elements is defined as the difference between the indices of those elements. For some applications which don't require the *distance* for detecting a *pattern*, e.g., traffic analysis, we can treat h as infinity.

Definition 1. A *pattern* is a sequence $\mathbf{Q} = q^{(1)}q^{(2)}\cdots q^{(l)}$, where $q^{(i)} \in \{0, 1, \cdots, r-1\}$ for any $i \in \{1, 2, \cdots, l\}$. A User u is said to have the *pattern* $\mathbf{Q}$ if

- The sequence $\mathbf{Q}$ is a subsequence (not necessarily consecutive) of user u's sequence (or its obfuscated sequence), and
- for $i \in \{1, 2, \cdots, l-1\}$, $q^{(i)}$ and $q^{(i+1)}$ appear in the sequence of user u (or its obfuscated sequence) with *distance* less than or equal to h.

Since we do not know which *pattern* the adversary might be using to identify a user, one proposed main idea is to ensure that the obfuscated sequence of each user includes a large number of potential *pattern*s. To achieve this, we define the concept of *superstring*s used in the obfuscation mechanism.

Definition 2. A sequence is an $(r, l)-$*superstring* if it contains all possible r^l length-l strings (repeated symbols allowed) on a size-r alphabet-$\mathcal{R}$ as its contiguous substrings (cyclic tail-to-head ligation not allowed here).

We define $f(r, l)$ as the length of the shortest $(r, l)-$*superstring*. A trivial upper bound on the length of a *superstring* on r symbols is lr^l, since this is the length of the $(r, l)-$*superstring* obtained by arranging all the possible r^l substrings without overlapping. As a result, we can conclude $f(r, l) \leq lr^l$. As an example of a superstring, the sequence 11221 is a $(2, 2)-$*superstring* because it contains 11, 12, 21, and 22 as its contiguous subsequences, thus $f(2, 2) \leq 5 \leq 8 = lr^l$. In fact, as we will see shortly, 5 is the lowest possible length of a $(2, 2)-$*superstring*, so $f(2, 2) = 5$.

Obfuscation Mechanism: Let $\mathbf{Z}_u$ be the $m \times 1$ vector containing the obfuscated version of user u's data sequence, and $\mathbf{Z}$ be the $m \times n$ matrix with the u^{th} column equal to $\mathbf{Z}_u$:

$$\mathbf{Z}_u = [Z_u(1), Z_u(2), \cdots, Z_u(m)]^T, \quad \mathbf{Z} = [\mathbf{Z}_1, \mathbf{Z}_2, \cdots, \mathbf{Z}_n].$$

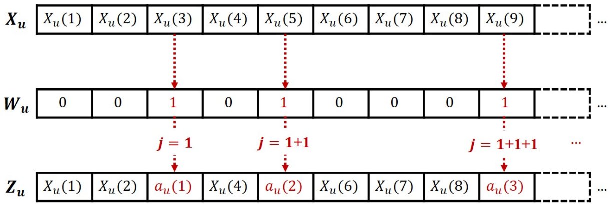
$\boldsymbol{X_u}$
$X_u(1)$ $X_u(2)$ $X_u(3)$ $X_u(4)$ $X_u(5)$ $X_u(6)$ $X_u(7)$ $X_u(8)$ $X_u(9)$ ...
$\boldsymbol{W_u}$
0 0 1 0 1 0 0 0 1 ...
$j = 1$
$j = 1+1$
$j = 1+1+1$
...
$\boldsymbol{Z_u}$
$X_u(1)$ $X_u(2)$ $a_u(1)$ $X_u(4)$ $a_u(2)$ $X_u(6)$ $X_u(7)$ $X_u(8)$ $a_u(3)$ ...

Fig. 2: The obfuscation of user $u \in \mathcal{U}$ based on an $(r, l)-$*superstring*

The obfuscation mechanism adds a noise to users' data in a way that the obfuscated data sequences are likely to have a large number of potential *pattern*s. This is to ensure that for any user and for any potential *pattern* that the adversary might obtain for that user, there will be a large number of other users with the same data *pattern* in their obfuscated data sequences.

The basic procedure is shown in Fig. 2. To create a noisy version of data samples, for each user we independently and randomly generate an $(r, l)-$*superstring* (one *superstring* is generated equally likely from the *superstring* solution set, explained in Section III), where r is the size of the data point alphabet, and l is the length of the *pattern*s. We denote the generated $(r, l)-$*superstring* as $\boldsymbol{a}_u = \{a_u(1), a_u(2), \cdots, a_u(L_s)\}$, where L_s is the length of the generated *superstring*. We define p_{ofb} as the probability that a user's data sample is changed to a different data sample by obfuscation. For each user at each specific time, we independently generate a random sequence variable $W_u(k)$ which has a Bernoulli distribution with parameter p_{ofb}. As shown in Fig. 2, the obfuscated version of the data sample of user u at time k can be written as:

$$Z_u(k) = \begin{cases} X_u(k), & \text{if } W_u(k) = 0 \\ a_u(j), & \text{if } W_u(k) = 1 \end{cases}$$

where $j = \sum\limits_{k'=1}^{k} W_u(k')$, and $a_u(j)$ is the j^{th} element of the $(r, l)-$*superstring* element used for the obfuscation. Note that if the length of the generated $(r, l)-$*superstring* is less than the length of the adversary's observed data sequences ($L_s \leq m$), we should append multiple copies of the generated *superstring* ($\mathbf{a}_u$) in order to have $L_s \geq m$.

Anonymization Mechanism: Anonymization is modeled by a random permutation Π on the set of n users, $\mathcal{U} = \{1, 2, \cdots, n\}$. Each user u is anonymized by the pseudonym function $\Pi(u)$. We use $\mathbf{Y}$ to denote the anonymized version of $\mathbf{Z}$; thus,

$$\begin{aligned} \mathbf{Y} &= \text{Perm}(\mathbf{Z}_1, \mathbf{Z}_2, \cdots, \mathbf{Z}_n; \Pi) \\ &= [\mathbf{Z}_{\Pi^{-1}(1)}, \mathbf{Z}_{\Pi^{-1}(2)}, \cdots, \mathbf{Z}_{\Pi^{-1}(n)}] \\ &= [\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_n], \end{aligned}$$

where $\text{Perm}(\ \cdot\ ; \Pi)$ is the permutation operation with permutation function Π. As a result, $\mathbf{Y}_u = \mathbf{Z}_{\Pi^{-1}(u)}$ and $\mathbf{Y}_{\Pi(u)} = \mathbf{Z}_u$. We employ the anonymization once in order to conceal the mapping between users and their data sequences.

Adversary Model: We assume an adversary who has access to length$-m$ observations per user; in other words, for each $u \in \{1, 2, \cdots, n\}$, the adversary observes $Y_{\Pi(u)}(1), Y_{\Pi(u)}(2), \cdots, Y_{\Pi(u)}(m)$. We also assume the adversary has identified a *pattern* $\mathbf{Q}_u$ of a specific user v, $q_v^{(1)} q_v^{(2)} \cdots q_v^{(l)}$, and is trying to uncover the user's identity based on the information that they have about the user's *pattern*. Note that we also assume that the adversary knows the obfuscation and the anonymization mechanisms; however, they do not know the realization of the generated *superstring* ($\mathbf{a}_u,\ u \in \mathcal{U}$) or the realization of the random permutation (Π).

We introduce the definition of $\epsilon-$*privacy* as:

Definition 3. User v with data *pattern* $q_v^{(1)} q_v^{(2)} \cdots q_v^{(l)}$ has $\epsilon-$*privacy* if for any other user u, the probability that user u has the same *pattern* as user v in their obfuscated data sequence is at least ϵ.

Loosely speaking, the above definition implies that even if the adversary can identify a data *pattern* of user v, they cannot identify user v with probability greater than $\frac{1}{n\epsilon}$. It is worth noting that this is a relatively strong requirement for privacy. In the common language of k-anonymity, this can be interpreted as $n\epsilon-$anonymity, as a specific user can be confused with $n\epsilon$ other users. This can be contrasted with the concept of perfect privacy [18], [24], [25], where it suffices that each user is confused with $N^{(n)}$ users, where $N^{(n)} \to \infty$ as $n \to \infty$. Hence, if we were to loosen the privacy definition, we can achieve privacy with lower obfuscation rates, and this investigation of utility-privacy tradeoffs is a future research direction.

III. Privacy Guarantee for Model-Free PPMs

Without loss of generality, consider $\epsilon-$privacy for an arbitrary user 1 with *pattern* sequence $q_1^{(1)} q_1^{(2)} \cdots q_1^{(l)}$. Per Section II, we want to ensure that the obfuscated data sequences or other users are likely to have the same data *pattern* as user 1 to confuse a pattern-matching adversary trying to find user 1. Let $\mathcal{B}_u$ be the event that user 1's *pattern* $q_1^{(1)} q_1^{(2)} \cdots q_1^{(l)}$ appears in user u's obfuscated data points $\mathbf{Z}_u$ (by Definition 1) due to our obfuscation technique. And here the *pattern* length l and the maximum *distance* h of the *pattern* letters appearing in the obfuscated sequence (by Definition 1) are assumed to be known and treated as constants.

We will assume a worst-case scenario: each user has their own unique *pattern* to be identified by the adversary. To be further pessimistic, let us start with the large value lr^l for the length of an $(r, l)-$*superstring*; we will show below that the length of the obfuscating *superstring* can be shortened by introducing the De Bruijn sequence. We will prove that such a *superstring* guarantees that at least a certain percentage ϵ of users will have the same *pattern* as user 1 after employing the obfuscation mechanism.

Definition 4. $\mathbb{P}(\mathcal{B}_u)$ is defined as the probability that the obfuscated sequence $\mathbf{Z}_u$ has user 1's identifying *pattern* due to obfuscation by an $(r, l)-$*superstring* with length lr^l (obtained by arranging all the possible r^l substrings without overlapping).

Definition 5. $\mathbb{P}(\mathcal{B}'_u)$ is defined as the probability that the obfuscated sequence $\mathbf{Z}_u$ has user 1's identifying *pattern* due to obfuscation by the shortest $(r, l)-$*superstring* with length $f(r, l)$.

Theorem 1. If $\mathbf{Z}$ is the obfuscated version of $\mathbf{X}$, and $\mathbf{Y}$ is the anonymized version of $\mathbf{Z}$ as defined previously, there exists a lower bound ϵ for the probability $\mathbb{P}(\mathcal{B}_u)$:

$$\mathbb{P}(\mathcal{B}_u) \geq \frac{\left(1 - (1 - p_{\text{obf}})^h\right)^{(l-1)}}{r^l} \sum_{\alpha=0}^{\min\left\{(r^l-1), \left\lfloor \frac{Gp_{\text{obf}}}{l} \right\rfloor\right\}} 1 - \exp\left(-\frac{\delta_\alpha^2}{2} G p_{\text{obf}}\right), \quad (1)$$

where

$$G = m - h(l-1), \qquad \delta_\alpha = 1 - \frac{\alpha l}{G p_{\text{obf}}} \text{ for } \alpha = 0, 1, \cdots, r^l - 1.$$

Proof. The procedure of obfuscation on any user $u \in \mathcal{U}$ is shown in Fig. 3. Note that since our generated *superstring* can have more than one copy of each *pattern*, we pessimistically focus on one "intended" copy of our desired *pattern* in the *superstring* for the analysis. We denote $L_{u,1}$ as the index of the first element of the "intended" version of the *pattern* in the *superstring*, such that $a_u(L_{u,1}) = q_1^{(1)}, a_u(L_{u,1}+1) = q_1^{(2)}, \cdots, a_u(L_{u,1}+l-1) = q_1^{(l)}$, and correspondingly, $M_{u,1}^{i}$ is the index of the data point $X_u(M_{u,1}^{(i)})$ that is obfuscated to $q_1^{(i)}$ $(M_{u,1}^{(i)} < m)$, for $i = 1, 2, \ldots, l$:

$$Z_u(M_{u,1}^{(i)}) = a_u(L_{u,1}+i-1) = q_1^{(i)}, \text{ for any } u \in \mathcal{U}. \quad (2)$$

According to Definition 1, the event $\mathcal{B}_u$ occurs if (but not only if) the following two events occur: (i) the user 1's *pattern* $q_1^{(1)}q_1^{(2)}\cdots q_1^{(l)}$ appears in user u's obfuscated data points $\mathbf{Z}_u$; and, (ii) the *distance* between any neighboring points of *pattern* $q_1^{(1)}q_1^{(2)}\cdots q_1^{(l)}$ in $\mathbf{Z}_u$ is smaller than or equal to h. Now, if we accordingly define event $\mathcal{E}_u$ and $\mathcal{F}_u$ as:

$$\mathcal{E}_u : M_{u,1}^{(1)} \leq m - h(l-1) = G, \quad (3)$$

$$\mathcal{F}_u : D_u^{(1)} \leq h; D_u^{(2)} \leq h; \cdots, D_u^{(l-1)} \leq h, \quad (4)$$

where $D_u^{(i)} = M_{u,1}^{(i+1)} - M_{u,1}^{(i)}$ are the *distance*s between $q_1^{(i+1)}$ and $q_1^{(i)}$ in user u's obfuscated sequence $\mathbf{Z}_u$, for $i = 1, 2, \ldots, l-1$. Since events $\mathcal{E}_u$ and $\mathcal{F}_u$ are independent, we have:

$$\mathbb{P}(\mathcal{B}_u) \geq \mathbb{P}(\mathcal{E}_u)\,\mathbb{P}(\mathcal{F}_u). \quad (5)$$

The probability of event $\mathcal{E}_u$ is the probability of $L_{u,1}$ successes in M Bernoulli trials, where each trial has probability of success p_{obf}; thus,

$$\mathbb{P}(\mathcal{E}_u) = \mathbb{P}(\text{at least } L_{u,1} \text{ success in } G \text{ trials}).$$

Since each user employs a randomly chosen superstring for obfusctation, the *pattern* is equally likely to be in any of the r^l substrings of length l; hence,

$$\mathbb{P}(L_{u,1} = \alpha l + 1) = \frac{1}{r^l}, \quad \alpha = 0, 1, \cdots, r^l - 1. \quad (6)$$

Thus, by employing the Law of Total Probability, we have:

$$\begin{aligned}\mathbb{P}(\mathcal{E}_u) = &\sum_{\alpha=0}^{r^l-1} \mathbb{P}\left(\text{at least } L_{u,1} \text{ success in } G \text{ trials}\middle| L_{u,1} = \alpha l + 1\right) \\ &\cdot \mathbb{P}(L_{u,1} = \alpha l + 1) \\ = &\frac{1}{r^l}\sum_{\alpha=0}^{r^l-1} \mathbb{P}(\text{at least } \alpha l + 1 \text{ success in } G \text{ trials}) \\ = &\frac{1}{r^l}\sum_{\alpha=0}^{r^l-1} \left[1 - \mathbb{P}(\text{less than } \alpha l + 1 \text{ success in } G \text{ trials})\right].\end{aligned}$$

Let us define $\mathcal{A}_\alpha$ as the event in which there exists less than $\alpha l + 1$ success in G trials. By employing the Chernoff Bound:

$$p(\mathcal{A}_\alpha) \leq \exp\left(-\frac{1}{2}\delta_\alpha^2 G p_{\text{obf}}\right), \quad \text{for all } \alpha < \frac{G p_{\text{obf}}}{l}. \quad (7)$$

Now, by using (6) and (7):

$$\mathbb{P}(\mathcal{E}_u) \geq \frac{1}{r^l} \sum_{\alpha=0}^{\min\left\{(r^l-1), \left\lfloor \frac{G p_{\text{obf}}}{l} \right\rfloor\right\}} 1 - \exp\left(-\frac{1}{2}\delta_\alpha^2 G p_{\text{obf}}\right). \quad (8)$$

Note that sub-events of $\mathcal{F}_u$: $D_u^{(1)} \leq h, \cdots, D_u^{(l-1)} \leq h$ are independent, thus, the probability of event $\mathcal{F}_u$ is:

$$\mathbb{P}(\mathcal{F}_u) = \prod_{i=1}^{l-1} \mathbb{P}\left(D_u^{(i)} \leq h\right) = \left(1 - (1 - p_{\text{obf}})^h\right)^{(l-1)}. \quad (9)$$

Thus, by (5), (8) and (9), we obtain (1). □

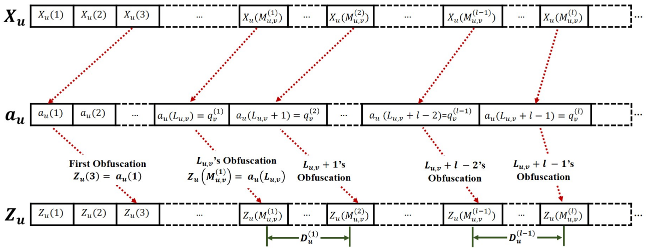

Fig. 3: The obfuscation of user $u \in \mathcal{U}$ for protecting its data trace from an adversary.

The methodology that we develop in Theorem 1 can readily be applied with $(r,l)-$*superstring*s with shorter length, from which we can provide stronger privacy guarantees. The following lemma provides a construction for the shortest $(r,l)-$*superstring* and evaluates its length.

Lemma 1. The length of a sequence solution for the shortest $(r,l)-$*superstring* is equal to r^l+l-1. That is, $f(r,l) = r^l+l-1$.

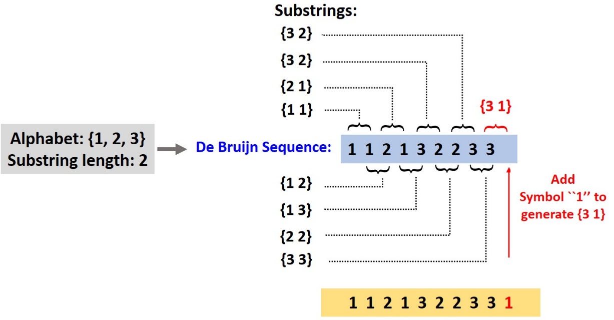

Fig. 4: The construction of a shortest $(3,2)-$*superstring* by using a De Bruijn sequence $B(3,2)$. The length of the constructed $(3,2)-$*superstring* is $f(3,2) = 3^2 + 2 - 1 = 10$.

Proof. A (non-unique) shortest $(r,l)-$*superstring* can be constructed by a De Bruijn sequence [38], [41], denoted by $B(r,l)$ in this paper. A De Bruijn sequence of order l on a size-r alphabet-$\mathcal{R}$, is a sequence with length r^l in which every possible length-l substring on $\mathcal{R}$ occurs exactly once as a contiguous subsequence. The last $(l-1)$ and the first $(l-1)$ letters of the De Bruijn sequence is cyclic tail-to-head ligation for counting the substrings. A shortest $(r,l)-$*superstring* can be constructed via one chosen De Bruijn sequence $B(r,l)$ by repeating $B(r,l)$'s front $(l-1)$ symbols at the end of the sequence, with length (r^l+l-1). We prove it in the following.

We first prove that the constructed sequence is an $(r,l)-$*superstring*. The sequence has the first $[r^l-(l-1)]$ substrings because it contains a full De Bruijn sequence $B(r,l)$ at its front r^l symbols. In addition, since the left $(l-1)$ substrings in $B(r,l)$ are counted by tracking from the last $(l-1)$ letters and the first $(l-1)$ letters as mentioned, the left $(l-1)$ substrings also appear in the constructed *superstring* in a non-cyclic way, since the De Bruijn sequence's front $(l-1)$ symbols have been copied to its end. Thus, the constructed sequence contains all possible r^l substrings, and hence, by Definition 2, it is a valid $(r,l)-$*superstring*.

Then we prove that the constructed sequence gives the shortest solution for an $(r,l)-$*superstring*. Each of these distinct substrings on the size-r alphabet-$\mathcal{R}$, must start at a different position in the sequence, because substrings starting at the same position are not distinct. Therefore, an $(r,l)-$*superstring* must have at least (r^l+l-1) symbols. Since the constructed $(r,l)-$*superstring* has exactly (r^l+l-1) symbols, it is optimally short with length (r^l+l-1). □

The solution for the shortest $(r,l)-$*superstring* is non-unique in general for $r \geq 2$ since we can construct our $(r,l)-$*superstring* by taking any De Bruijn sequence $B(r,l)$ (which is also non-unique: another De Bruijn sequence can be generated by circular shifting $B(r,l)$ in the left or right direction by some digits) from any De Bruijn sequence pattern set. An example of construction of a shortest $(3,2)-$*superstring* by a De Bruijn sequence $B(3,2)$ = "112132233" is shown in Fig. 4, and its length $f(3,2) = 10$. Another solution can be generated, for instance, by right circular shifting the De Bruijn sequence $B(3,2)$ by one digit and copy the first symbol '3' to its end: "3112132233". Another solution can be: "3311213223" (by further right circular shifting $B(3,2)$ by one digit and adding its front symbol '3' to its end).

Next we consider the privacy performance when the shortest $(r,l)-$*superstring* is employed.

Theorem 2. If $\mathbf{Z}$ is the obfuscated version of $\mathbf{X}$, and $\mathbf{Y}$ is the anonymized version of $\mathbf{Z}$ as defined previously, there exists a lower bound ϵ' for the probability $\mathbb{P}\left(\mathcal{B}'_u\right)$:

$$\mathbb{P}\left(\mathcal{B}'_u\right) \geq \frac{\left(1-(1-p_{\text{obf}})^h\right)^{(l-1)}}{r^l} \sum_{\alpha=0}^{\min\left\{(r^l-1),\, \lfloor G p_{\text{obf}} \rfloor\right\}} 1-\exp\left(-\frac{\delta'^2_\alpha}{2} G p_{\text{obf}}\right), \quad (10)$$

where

$$G = m - h(l-1), \quad \delta'_\alpha = 1 - \frac{\alpha}{G p_{\text{obf}}}, \text{ for } \alpha = 0, 1, \cdots, r^l - 1.$$

Proof. By using (5), we have:

$$\mathbb{P}\left(\mathcal{B}'_u\right) \geq \mathbb{P}\left(\mathcal{E}'_u\right) \mathbb{P}\left(\mathcal{F}'_u\right), \quad (11)$$

where the events $\mathcal{E}'_u$ and $\mathcal{F}'_u$ are defined analogously to the events $\mathcal{E}_u$ and $\mathcal{F}_u$ defined in (3) and (4), respectively.

For a given *superstring* set generated by a De Bruijn sequence $B(r,l)$, we assume that the index values $L_{u,1}$ are equally likely over the front r^l indices in the $(r,l)-$*superstring* chosen by user u since one $(r,l)-$*superstring* can be selected by uniformly circular shifting $B(r,l)$ by Lemma 1. So we have:

$$\mathbb{P}\left(L_{u,1} = \alpha + 1\right) = \frac{1}{r^l}, \quad \alpha = 0, 1, \cdots, r^l - 1. \quad (12)$$

Similarly, by employing a Chernoff Bound and the Law of Total Probability, we have:

$$\mathbb{P}\left(\mathcal{E}'_u\right) \geq \frac{1}{r^l} \sum_{\alpha=0}^{\min\left\{(r^l-1),\, \lfloor G p_{\text{obf}} \rfloor\right\}} 1-\exp\left(-\frac{1}{2}\delta'^2_\alpha G p_{\text{obf}}\right). \quad (13)$$

In addition, the probability of event $\mathcal{F}'_u$ can be obtained similarly by (9):

$$\mathbb{P}\left(\mathcal{F}'_u\right) = \mathbb{P}\left(\mathcal{F}_u\right). \quad (14)$$

Therefore, by (11), (13), and (14), we obtain (10). □

Discussion: In Table I, we show the lower bounds to ϵ and ϵ' for different parameter settings. These results show that our PPMs will yield a percentage of the user set $\mathcal{U}$ that have the same *pattern* in their obfuscated sequences as user 1's; Increasing m or p_{obf} increases the chance that other users have the same *pattern* as user 1's.

TABLE I: Numerical evaluation of the lower bounds ϵ and ϵ' for $\mathbb{P}\left(\mathcal{B}_u\right)$ and $\mathbb{P}\left(\mathcal{B}'_u\right)$ for different parameter settings.

m	r	l	h	p_{obf}	lower bound ϵ	lower bound ϵ'
1000	20	3	8	10%	0.11%	0.35%
1000	20	3	10	15%	0.35%	1.06%
3000	16	2	10	10%	35.40%	65.06%
3000	16	2	10	15%	66.34%	80.31%
5000	30	2	10	10%	17.07%	34.12%
10000	30	2	10	10%	34.75%	65.12%
15000	12	3	10	10%	11.87%	35.59%
15000	15	3	10	10%	6.07%	18.22%

IV. Conclusion

The need for sharing sensitive data in today's interconnected world has led to major privacy and security concerns for users; thus, different privacy-preserving mechanisms (PPMs) have been proposed to improve users' privacy. A key parameter in the design of many PPMs is a statistical model of users' data. However, if the modeling assumptions are not accurate, privacy guarantees are no longer valid. Unlike prior work in this area, here we make no specific assumptions about the statistical model of the users and propose a PPM to achieve privacy guarantees by applying small artificial distortion to thwart pattern matching attacks. In particular, a small noise has been added to users' data in a way that the obfuscated data sequences are likely to have a large number of potential patterns; thus, for any user and for any potential pattern that the adversary might have to identify that user, we have shown that there will be a large number of other users with the same data pattern in their obfuscated data sequences.

References

[1] F. Staff, "Internet of things: Privacy and security in a connected world," *Technical report, Federal Trade Commission*, 2015.

[2] P. Porambage, M. Ylianttila, C. Schmitt, P. Kumar, A. Gurtov, and A. V. Vasilakos, "The quest for privacy in the internet of things," *IEEE Cloud Computing*, vol. 3, no. 2, pp. 36–45, 2016.

[3] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial internet of things," in *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. IEEE, 2015, pp. 1–6.

[4] H. Wang and F. P. Calmon, "An estimation-theoretic view of privacy," in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2017, pp. 886–893.

[5] A. Ukil, S. Bandyopadhyay, and A. Pal, "Iot-privacy: To be private or not to be private," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2014, pp. 123–124.

[6] S. Hosseinzadeh, S. Rauti, S. Hyrynsalmi, and V. Leppänen, "Security in the internet of things through obfuscation and diversification," in *IEEE Conference on Computing, Communication and Security (ICCCS)*. Pamplemousses, Mauritius: IEEE, 2015, pp. 1–5.

[7] H. Hsu, S. Asoodeh, and F. P. Calmon, "Information-theoretic privacy watchdogs," in *2019 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2019, pp. 552–556.

[8] J. Ren, D. J. Dubois, D. Choffnes, A. M. Mandalari, R. Kolcun, and H. Haddadi, "Information exposure from consumer iot devices: A multidimensional, network-informed measurement approach," in *Proceedings of the Internet Measurement Conference*, 2019, pp. 267–279.

[9] S. Sreekumar and D. Gündüz, "Optimal privacy-utility trade-off under a rate constraint," in *2019 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2019, pp. 2159–2163.

[10] H. Wang, L. Vo, F. P. Calmon, M. Médard, K. R. Duffy, and M. Varia, "Privacy with estimation guarantees," *IEEE Transactions on Information Theory*, vol. 65, no. 12, pp. 8025–8042, 2019.

[11] M. Diaz, H. Wang, F. P. Calmon, and L. Sankar, "On the robustness of information-theoretic privacy measures and mechanisms," *IEEE Transactions on Information Theory*, vol. 66, no. 4, pp. 1949–1978, 2019.

[12] F. Shirani, S. Garg, and E. Erkip, "Optimal active social network deanonymization using information thresholds," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 1445–1449.

[13] B. Hoh and M. Gruteser, "Protecting location privacy through path confusion," in *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm)*. Pamplemousses, Mauritius: IEEE, 2005, pp. 194–205.

[14] C. L. Claiborne, C. Ncube, and R. Dantu, "Random anonymization of mobile sensor data: Modified android framework," in *2015 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE, 2015, pp. 182–184.

[15] J. Unnikrishnan, "Asymptotically optimal matching of multiple sequences to source distributions and training sequences," *IEEE Transactions on Information Theory*, vol. 61, no. 1, pp. 452–468, 2014.

[16] K. Sung, J. Biswas, E. Learned-Miller, B. N. Levine, and M. Liberatore, "Server-side traffic analysis reveals mobile location information over the internet," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1407–1418, 2018.

[17] F. M. Naini, J. Unnikrishnan, P. Thiran, and M. Vetterli, "Where you are is who you are: User identification by matching statistics," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 358–372, 2016.

[18] Z. Montazeri, A. Houmansadr, and H. Pishro-Nik, "Achieving Perfect Location Privacy in Wireless Devices Using Anonymization," *IEEE Transaction on Information Forensics and Security*, vol. 12, no. 11, pp. 2683–2698, 2017.

[19] Y. Yoshida, M.-H. Yung, and M. Hayashi, "Optimal mechanism for randomized responses under universally composable security measure," in *2019 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2019, pp. 547–551.

[20] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st international conference on Mobile systems, applications and services*. San Francisco, California, USA: ACM, 2003.

[21] C. A. Ardagna, M. Cremonini, E. Damiani, S. D. C. di Vimercati, and P. Samarati, "Location privacy protection through obfuscation-based techniques," in *DBSec*, 2007.

[22] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *EUROCRYPT*, 2006.

[23] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 63–69, 1965.

[24] N. Takbiri, A. Houmansadr, D. L. Goeckel, and H. Pishro-Nik, "Matching anonymized and obfuscated time series to users' profiles," *IEEE Transactions on Information Theory*, vol. 65, no. 2, pp. 724–741, 2019.

[25] N. Takbiri, A. Houmansadr, D. L. Goeckel, and H. Pishro-Nik, "Limits of locatin privacy under anonymization and obfuscation," in *International Symposium on Information Theory (ISIT)*. Aachen, Germany: IEEE, 2017, pp. 764–768.

[26] S. Mangold and S. Kyriazakos, "Applying pattern recognition techniques based on hidden markov models for vehicular position location in cellular networks," in *Gateway to 21st Century Communications Village. VTC 1999-Fall. IEEE VTS 50th Vehicular Technology Conference (Cat. No. 99CH36324)*, vol. 2. IEEE, 1999, pp. 780–784.

[27] B.-H. Juang and L. R. Rabiner, "The segmental k-means algorithm for estimating parameters of hidden markov models," *IEEE Transactions on acoustics, speech, and signal Processing*, vol. 38, no. 9, pp. 1639–1641, 1990.

[28] F. Shirani, S. Gar, and E. Erkip, "A concentration of measure approach to database de-anonymization," in *IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2019.

[29] D. Cullina, P. Mittal, and N. Kiyavash, "Fundamental limits of database alignment," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 651–655.

[30] O. E. Dai, D. Cullina, and N. Kiyavash, "Fundamental limits of database alignment," in *Proceedings of Machine Learning Research,*, 2019.

[31] N. Takbiri, R. Soltani, D. Goeckel, A. Houmansadr, and H. Pishro-Nik, "Asymptotic loss in privacy due to dependency in gaussian traces," in *IEEE Wireless Communications and Networking Conference (WCNC)*. Marrakech, Morocco: IEEE, 2019.

[32] B. Wang, W. Song, W. Lou, and Y. T. Hou, "Privacy-preserving pattern matching over encrypted genetic data in cloud computing," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[33] M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama, and T. Koshiba, "Secure pattern matching using somewhat homomorphic encryption," in *Proceedings of the 2013 ACM workshop on Cloud computing security workshop*. ACM, 2013, pp. 65–76.

[34] M. Newey, "Notes on a problem involving permutations as subsequences," STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, Tech. Rep., 1973.

[35] S. Radomirovic, "A construction of short sequences containing all permutations of a set as subsequences," *The Electronic Journal of Combinatorics*, vol. 19, no. 4, p. 31, 2012.

[36] N. Johnston, "The minimal superpermutation problem," http://www.njohnston.ca/2013/04/the-minimal-superpermutation-problem/, accessed: 2020-01-15.

[37] R. Houston, "Tackling the minimal superpermutation problem," *arXiv preprint arXiv:1408.5108*, 2014.

[38] N. Bruijn, de, "A combinatorial problem," *Proceedings of the Section of Sciences of the Koninklijke Nederlandse Akademie van Wetenschappen te Amsterdam*, vol. 49, no. 7, pp. 758–764, 1946.

[39] N. Takbiri, A. Houmansadr, D. L. Goeckel, and H. Pishro-Nik, "Privacy against statistical matching: Inter-user correlation," in *International Symposium on Information Theory (ISIT)*. Vail, Colorado, USA: IEEE, 2018, pp. 1036–1040.

[40] N. Takbiri, A. Houmansadr, D. Goeckel, and H. Pishro-Nik, "Privacy of dependent users against statistical matching," *IEEE Transactions on Information Theory*, 2020.

[41] F. S. Annexstein, "Generating de bruijn sequences: An efficient implementation," *IEEE Transactions on Computers*, vol. 46, no. 2, pp. 198–200, 1997.