Superstring-Based Sequence Obfuscation to Thwart Pattern Matching Attacks

Bo Guan, *Student Member, IEEE*, Nazanin Takbiri, *Student Member, IEEE*, Dennis Goeckel, *Fellow, IEEE*, Amir Houmansadr, *Member, IEEE*, and Hossein Pishro-Nik, *Member, IEEE*

***Abstract*—User privacy can be compromised by matching user data traces to records of their previous behavior. The matching of the statistical characteristics of traces to prior user behavior has been widely studied. However, an adversary can also identify a user deterministically by searching data traces for a pattern that is unique to that user. Our goal is to thwart such an adversary by applying small artificial distortions to data traces such that each potentially identifying pattern is shared by a large number of users. Importantly, in contrast to statistical approaches, we develop data-independent algorithms that require no assumptions on the model by which the traces are generated. By relating the problem to a set of combinatorial questions on sequence construction, we are able to provide provable guarantees for our proposed constructions. We also introduce data-dependent approaches for the same problem. The proposed obfuscation methods are evaluated on synthetic data traces and on the Reality Mining Data set to demonstrate the performance of the proposed algorithms relative to alternatives.**

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

I. Introduction

THE PROMINENCE of the Internet of Things (IoT) has raised security and privacy concerns. The problem considered here addresses several important scenarios: fingerprinting webpages visited by users through anonymous communication systems [2], [3], linking communicating parties on messaging applications [4], and inferring the activities of the users of IoT devices [5], [6]. While the setting is general, we motivate the problem from the consideration of user-data driven (UDD) services in IoT applications: data submitted by users is analyzed to improve service in applications, such as health care, smart homes, and connected vehicles. But privacy and security threats are a major obstacle to the wide adoption of IoT applications [7], [8], [9], [10], [11]. Often anonymization and obfuscation mechanisms are proposed to improve privacy at the cost of user utility. Anonymization techniques frequently change the pseudonym of users [2], [12], [13], [14], [15], [16], whereas obfuscation techniques add noise to users' data samples [17], [18], [19], [20], [21], [22], [23].

Privacy can be compromised by linking the characteristics of a target sequence of activities to previously observed user behavior. To provide privacy guarantees in the presence of such potential *sequence matching*, a stochastic model for the users' data (e.g., Markov chains) has been generally assumed [24], [25], [26], [27], [28], [29], and privacy attacks that match the statistical characteristics of the target sequence to those of past sequences of the user are considered. These previous approaches have two limitations: 1) many privacy attacks are based on simple "pattern matching" for identification [14], classification [30], [31], or prediction [32], where the adversary (algorithm) looks (deterministically) for a specific ordered sequence of values in the user's data and 2) as privacy-protection mechanism (PPM) designers, we may not know the underlying statistical model for users' data. In particular, Takbiri *et al.* [22] have shown that modeling errors can destroy privacy guarantees.

We consider the following important question: Can we thwart privacy attacks that de-anonymize users by finding specific identifying patterns in their data, *even if we do not know what patterns the adversary might be exploiting*, and can we do so without assuming a certain model (or collection of models) for users' data? Our privacy metric and the resulting obfuscation approach are based on the following idea: *noise should be added in a way that the obfuscated user 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 be a large number of other users with the same data pattern in their obfuscated sequences. By focusing on this common type of privacy attack (pattern matching), the PPM is able to eliminate the need for making specific assumptions about the users' data model.

To achieve privacy guarantees, we first introduce a data-independent obfuscation (DIO) approach, which means that sequence obfuscation can be performed without knowledge of the actual data values in a user's sequence. A DIO mechanism can be even deployed on the traffic of oblivious users, e.g., it can be run on an IoT gateway/router to apply obfuscations on transit IoT traffic. Such data independence would be of value in various applications where the upcoming events

Manuscript received 17 June 2021; revised 3 March 2022; accepted 18 August 2022. Date of publication 5 September 2022; date of current version 21 November 2022. This work was supported by the National Science Foundation under Grant CNS–1739462. This article was presented in part at IEEE International Symposium on Information Theory, Los Angeles, CA, USA, 21–26 June 2020. [DOI: 10.1109/ISIT44484.2020.9174069]. *(Corresponding author: Bo Guan.)*

Bo Guan, Dennis Goeckel, and Hossein Pishro-Nik are with the Department of Electrical and Computer Engineering, University of Massachusetts at Amherst, Amherst, MA 01003 USA (e-mail: boguan@ecs.umass.edu; goeckel@ecs.umass.edu; pishro@ecs.umass.edu).

Nazanin Takbiri is with Microsoft Corporation, Mountain View, CA 94043 USA (e-mail: ntakbiri@ecs.umass.edu).

Amir Houmansadr is with the Manning College of Information and Computer Sciences, University of Massachusetts at Amherst, Amherst, MA 01003 USA (e-mail: amir@cs.umass.edu).

Digital Object Identifier 10.1109/JIOT.2022.3203995

2327-4662 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on December 15,2022 at 17:51:05 UTC from IEEE Xplore. Restrictions apply.

are not known *a priori*, for instance in website fingerprinting [2], [3] and flow correlation [33] applications where obfuscations need to be applied on live (nonbuffered) network packets. Our approach relies on the concept of *superstring*s, which contain every possible pattern of length less than or equal to the pattern length l (repeated symbols are allowed in each contiguous substring). This in turn happens to be related to a rich area in combinatorics [34], [35], [36], [37]. Since the DIO approach does not take advantage of the data values which appear in users' previous obfuscated sequence, its performance (fraction of users having the same pattern) would be largely decided by the obfuscation sequence noise. After introducing and characterizing our DIO approach, we introduce data-dependent obfuscation (DDO) approaches for comparison; data-dependent approaches are able to look at the values in the users' data sequences and base their obfuscation on such information. A DDO mechanism should be used in scenarios where the data generator (e.g., IoT devices, or users) are helping with the application of the obfuscation mechanism (i.e., by integrating the obfuscation software into their system). Such data dependence would be possible in applications where the whole vector of user data is known to the obfuscation party at the obfuscation time, for instance image processing applications [38], [39], [40]. The DDO techniques are able to utilize the statistics regarding patterns in the previous obfuscated sequence, which means they could potentially have better performance than the data-independent approach, as illustrated in numerical results of Section VIII. However, the DDO requires much more information than the data-independent approaches and hence has more time complexity overhead, as illustrated in complexity analysis in Section VII.

But designing such defenses to thwart pattern matching attacks is a challenging problem: we do not usually know what identifying patterns will be appearing in users' data sequences, hence motivating the challenging goal of providing a systematic defense mechanism which is able to ensure that all of the patterns appear in a large number of users' data sequences. As will be observed from our analysis, the mathematical modeling of the proposed algorithm and its achievable privacy performance is also a challenging task.

The contributions of our paper are listed.

1) We propose a formal framework for defending against pattern matching attacks when there is no statistical model for the user data (Section III).
2) We present a DIO approach based on *superstring*s and by lower bounding its performance for two different types of *superstring*s, prove that it yields a nonzero fraction of user sequences that contain a potentially identifying pattern (Section IV).
3) We develop DDO approaches for our pattern matching framework (Section VI).
4) We validate the developed approaches and state-of-the-art alternatives on both synthetic data and the Reality Mining data set to demonstrate their utility and compare their performance (Section VIII).

Finally, we present the conclusions that can be drawn from our study in Section IX.

II. Related Work

IoT devices provide important services but they have multiple potential privacy issues: 1) enabling unauthorized access and misuse of personal information [10], [41]; 2) facilitating attacks on other systems [7], [8]; and 3) creating personal privacy and safety problems [42], [43], [44].

In recent years, anonymization techniques have been studied, which conceal the mapping between users' identity and data by periodically changing the mapping to prevent statistical inference attacks. The k-anonymity protection approach is proposed in [45] and [46], which guarantees that the information for any person contained in the released version of the data cannot be distinguished from at least $k-1$ individuals. In [47], [48], and [49], k-anonymity is adopted for enhancing the privacy of moving objects. However, finding optimal k-anonymization (by generalization) is NP-hard and those methods' performances are affected by their prior knowledge about users' data sequences; moreover, few work focuses on protecting privacy of objects which could be potentially identified by adversaries based on their moving pattern (or subsequence) as their quasi-identifier (QID). Besides, there is little work which systematically designs defense mechanisms against pattern matching attacks and mathematically models the obfuscation process and its achievable provable privacy. In our work, we propose privacy-protecting algorithms for preventing users' profiles being identified by linking their identifying patterns. We provide an obfuscation solution (data-independent) which is able to create plenty of unique patterns for a large number of users without sacrificing significant utility. We categorize the relevant existing works for privacy-protecting mechanisms into the following types, which might potentially reduce the risk of being identified by pattern matching: 1) generalization [45], [50], [51], [52], [53], [54]; 2) subsampling [55], [56], [57], [58]; and 3) obfuscation [22], [59], [60], [61]. The generalization-based method prevents data sequences being identified by reducing the resolution at which data is reported. For instance, ZIP codes can be generalized by dropping the least significant (rightmost) digit at each generalization step [45], [50], [51]. Due to the degrading of the data point resolution, the generalization method can potentially increase the probability that users share a given potentially identifying sequence. The data subsampling technique, which reduces the sampling frequency, might potentially lower the identification risks by avoiding the presence of identifying subsets of the data points [55], [56], [57], [58]. Obfuscation techniques increase data privacy by adding noises to the data sequences, such as uniform noise [22], [60].

Anonymization is often insufficient because an adversary can track users' identities by using users' trajectory information [12], [14] or specific patterns [15], [62], [63]. Privacy preservation could be challenged by de-anonymization attacks if some sensitive information is known by an adversary [13], [64], [65]. Hence, obfuscation techniques protect users' privacy by introducing perturbations into users' data sequences to decrease their accuracy [59]. Gruteser and Grunwald [18] proposed an adaptive algorithm which adjusts the resolution of location information along spatial

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on December 15,2022 at 17:51:05 UTC from IEEE Xplore. Restrictions apply.

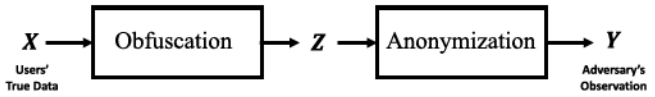

Fig. 1. Applying obfuscation and anonymization techniques to the users' data points.

and temporal dimensions. In [19], a comprehensive solution aimed at preserving location privacy of individuals through artificial perturbations of location information is presented. The work of [20] provides efficient distributed protocols for generating random noise to provide security against malicious participants. In [21], a randomized response method is proposed which allows interviewees to maintain privacy while increasing cooperation.

Data protection mechanisms might limit the utility when data also needs to be shared with an application or the provider to achieve some utility [9] or a quality of service constraint [12]. Theoretical analyses of the privacy-utility tradeoff (PUT) are provided in [11] and [17]. A key concept of *relevance* is proposed which strikes a balance between the need of service providers, requiring a certain level of location accuracy, and the need of users, asking to minimize the disclosure of personal location information [19]. Takbiri *et al.* [22], [23] derived the theoretical bounds on the privacy versus utility of users when an adversary is trying to perform statistical analyses on time series to match the series to user identity.

Pattern matching problems have attracted researchers in recent years, in particular fast pattern matching [62], [63], database search [66], [67], [68], secure pattern matching [69], [70], [71], and identification [1], [14], [30], [72] and characterization [73], [74], [75], [76] by using patterns or subsequences.

III. System Model, Definitions, and Metrics

Consider a system with n users whose identification we seek to protect. Let $X_u(k)$ denote the data of user u at time k. We assume there are $r \geq 2$ possible values for each of the users' data points in a finite size set $\mathcal{R} = \{0, 1, \ldots, r-1\}$. 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 uth column equal to $\mathbf{X}_u$

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

As shown in Fig. 1, in order to achieve privacy for users, both anonymization and obfuscation techniques are employed. In Fig. 1, $\mathbf{Z}$ denotes the reported data of the users after applying the obfuscation, and $\mathbf{Y}$ denotes the reported data after applying the obfuscation and the anonymization, where, with $Z_u(k)$ denoting the obfuscated data of user u at time k and $Y_u(k)$ denoting the obfuscated and anonymized data of user u at time k, respectively, $\mathbf{Z}$ and $\mathbf{Y}$ are defined analogously to $\mathbf{X}$.

Next, we provide a formal definition of a *pattern*. As an example, a potential pattern could be the sequence of locations that the user normally visits in a particular order: their office, the gym, a child's school. 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 within *distance* less than or equal to h, where the *distance* between two elements is defined as the difference between the indices of those elements ($h \geq 1$). The parameter h could have value one for the most restricted case: the elements of the pattern sequence must appear consecutively in users' sequences. And, h could be infinity for the unconstrained case: applications which do not consider *distance* for detecting a pattern, e.g., traffic analysis.

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

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

Obfuscation Mechanism: Given the model above and the definition of ϵ-*privacy* below, the objective is to design obfuscation schemes for the user data sequences that maximize ϵ-*privacy* with minimum sequence distortion without knowing what pattern the adversary might be exploiting or which user the adversary might be targeting. For simplicity, we consider the case of sparsely sampled data and thus leave to future work the enforcement of consistency constraints on the obfuscated user data sequences, such as a continuity constraint requiring that adjacent sequence elements have similar values. The design and characterization of obfuscation mechanisms is the main topic of the succeeding sections.

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

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

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 $\mathbf{Y}_{\Pi(u)} = \mathbf{Z}_u$. In practice, our model would arise when anonymization takes place every m samples. In such a case, it is sufficient to assume sequences of length m and employ the anonymization once to conceal the mapping between users and their data sequences. We assume the anonymization process is in a uniform manner [22], [60], [77], as in previous work, which means each mapping of the profile anonymization for each user is equally likely. This is optimal and readily achieved.

Adversary Model: The adversary has access to a sequence of observations of length m for each user; in other words, for each $u \in \{1, 2, \ldots, n\}$, the adversary observes $Y_{\Pi(u)}(1), Y_{\Pi(u)}(2), \ldots, Y_{\Pi(u)}(m)$. We also assume the adversary has identified a *pattern* $\mathbf{Q}_v$ of a specific user v, $q_v^{(1)}q_v^{(2)}, \ldots, q_v^{(l)}$, and is trying to identify the sequence of a user v by finding the sequence with pattern $q_v^{(1)}q_v^{(2)}, \ldots, q_v^{(l)}$. The adversary knows the obfuscation and the anonymization mechanisms; however, they do not know the realization of the random permutation (Π) and they do not know the

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on December 15,2022 at 17:51:05 UTC from IEEE Xplore. Restrictions apply.

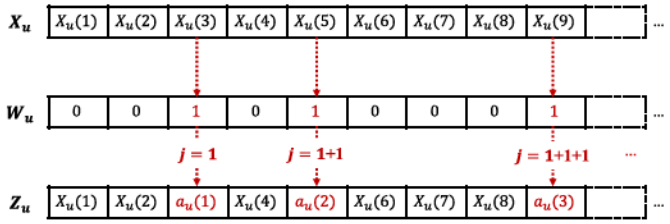

Fig. 2. Obfuscation of the data sequence of user $u \in \mathcal{U}$ based on an (r, l)-*superstring*.

realization of any randomly generated elements of the obfuscation mechanism.

We define ϵ-*privacy* as follows.

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

Loosely speaking, this implies that the adversary cannot identify user v with probability better than $(1/n\epsilon)$. If we assume ϵ is a constant independent of n, ϵ-privacy is a strong requirement for privacy—equivalent to $n\epsilon$-anonymity in the setting of k-anonymity. In contrast, in perfect privacy [16], [22], [23] it suffices that each user is confused with $N^{(n)}$ users, where $N^{(n)} \to \infty$ as $n \to \infty$. Hence, we will also consider cases where ϵ is a decreasing function of n so as to consider less stringent privacy definitions.

IV. Privacy Guarantee for Model-Free PPMs

We present constructions for model-free privacy-protection mechanisms under the model of Section III and then characterize their performance.

A. Constructions

For any user and for any potential pattern that the adversary might obtain for that user, we want to ensure there will be a large number of other users with the same data pattern in their obfuscated data sequences. First, we define the concept of a *superstring* and then our obfuscation mechanism.

Definition 3: 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).

We define $f(r, l)$ as the length of the shortest (r, l)-*superstring*. A trivial upper bound is $f(r, l) \leq lr^l$, as lr^l is the length of the (r, l)-*superstring* obtained by concatenating all possible r^l substrings. 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 $f(2, 2) \leq 5 \leq 8 = lr^l$.

Superstring-Based Obfuscation (SBU): Recall that $\mathbf{Z}_u$ is the $m \times 1$ vector of the obfuscated version of user u's data sequence, and $\mathbf{Z}$ is the $m \times n$ matrix with uth column $\mathbf{Z}_u$

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

The basic procedure is shown in Fig. 2. For each user, we independently and randomly generate an (r, l)-*superstring* from the *superstring* solution set described below. We denote the generated (r, l)-*superstring* as $\boldsymbol{a}_u = \{a_u(1), a_u(2), \ldots, a_u(L_s)\}$, where L_s is the length of the generated *superstring*. The parameter p_{obf} is the probability that we will change a given data sample. Thus, for each data point of each user, we independently generate a Bernoulli random variable $W_u(k)$ with parameter p_{obf}. As shown in Fig. 2, the obfuscated version of the data sample of user u at time k can then be written as follows:

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

where $j = \sum_{k'=1}^{k} W_u(k')$, and $a_u(j)$ is the jth element of the (r, l)-*superstring* used for the obfuscation. If the length of the generated (r, l)-*superstring* is not sufficient (i.e., $\sum_{k'=1}^{m} W_u(k') > L_s$), we choose another *superstring* at random to continue.

Independent and Identically Distributed (i.i.d.) Obfuscation: In [22], [60], and [78] a *uniform i.i.d.* obfuscation mechanism is used. For each user, an i.i.d. sequence of random variables $\mathbf{b_u} = \{b_u(1), b_u(2), \ldots, \}$ uniformly distributed on the alphabet $\mathcal{R} = \{0, 1, \ldots, r-1\}$ is generated. These values are used to obfuscate the sequence $X_u(k)$: for each data point of each user, we independently generate a Bernoulli random variable $W_u(k)$ with parameter p_{obf}. The obfuscated version of the data sample of user u at time k can be written as follows:

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

where $j = \sum_{k'=1}^{k} W_u(k')$. The i.i.d. obfuscation will be a benchmark for comparison of our superstring-based approaches.

B. Analysis

Without loss of generality, consider ϵ-privacy for user 1 with pattern sequence $q_1^{(1)} q_1^{(2)}, \ldots, q_1^{(l)}$. The pattern length l and the maximum *distance* h between the appearance of pattern elements are assumed to be known and treated as constants, but we hasten to note that this defends against an attacker employing a pattern with a length less than or equal to l and maximum *distance* greater than or equal to h.

We assume a worst-case scenario: user 1 has a pattern unique to their data set that can be exploited for identification. We start with the upper bound lr^l for the length of an (r, l)-*superstring*. We will prove that such a *superstring* guarantees that at least a certain fraction ϵ of users will have the same pattern as user 1 after employing the obfuscation mechanism. Later, we will improve this result by introducing the De Bruijn sequence to shorten the *superstring*.

Definition 4: Let $\mathcal{B}_u$ be the event 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 concatenating all possible r^l substrings.

Theorem 1: The probability of $\mathcal{B}_u$, denoted by $\mathbb{P}(\mathcal{B}_u)$, is lower bounded by a constant that does not depend on n as

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on December 15,2022 at 17:51:05 UTC from IEEE Xplore. Restrictions apply.

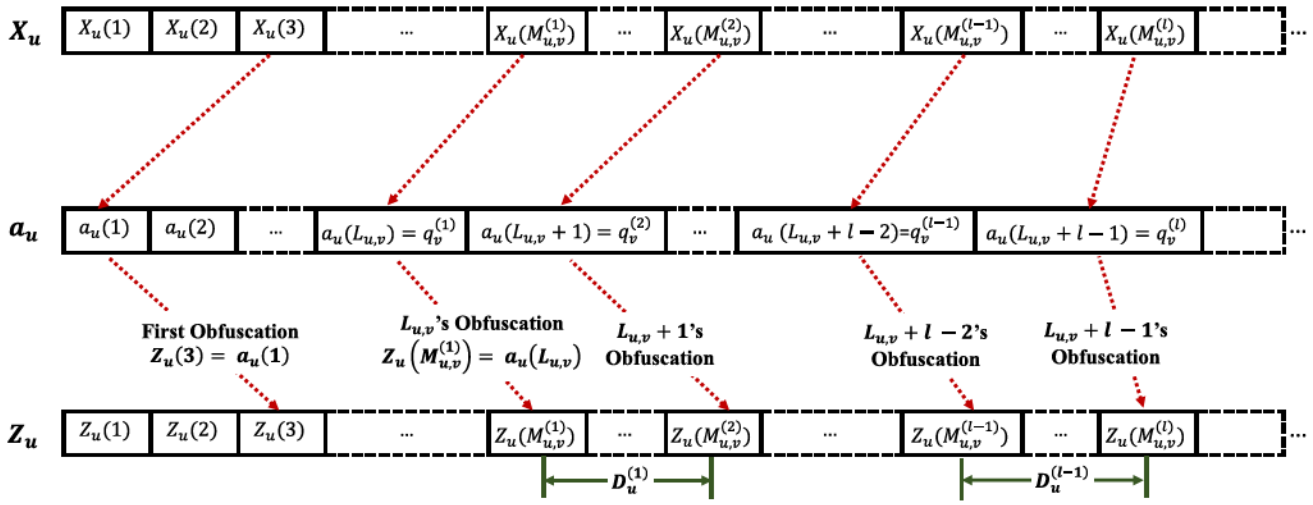

Fig. 3. Notation for the proof of Theorem 1 in the obfuscation of the trace of user $u \in \mathcal{U}$.

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

where

$$G = m - h(l-1),\ \delta_\alpha = 1 - \frac{\alpha l}{Gp_{\text{obf}}},\quad \text{for } \alpha = 0, 1, \ldots, r^l - 1$$

Proof: The notation employed here within the procedure of obfuscation for user $u \in \mathcal{U}$ is shown in Fig. 3. Note that our generated *superstring* can have more than one copy of each pattern, but we pessimistically focus on one copy of our desired pattern. We denote $L_{u,1}$ as the index of the first element of the pattern of the *superstring* of user u, such that $a_u(L_{u,1}) = q_1^{(1)}, a_u(L_{u,1}+1) = q_1^{(2)}, \ldots, a_u(L_{u,1}+l-1) = q_1^{(l)}$, 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 $i = 1, 2, \ldots, l$

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

The sequences X_u and Z_u can be assumed to be infinitely long with the adversary only seeing the first m elements of Z_u. Therefore, a sufficient condition for $\mathcal{B}_u$ (according to Definition 1) is $\mathcal{E}_u \bigcap \mathcal{F}_u$, where

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

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

where $D_u^{(i)} = M_{u,1}^{(i+1)} - M_{u,1}^{(i)}$ are the *distances* between $q_1^{(i+1)}$ and $q_1^{(i)}$ in user u's obfuscated sequence $\boldsymbol{Z}_u$, for $i = 1, 2, \ldots, l-1$. Note that we have defined $\mathcal{E}_u$ and $\mathcal{F}_u$ so as to make them independent. Thus, we have

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

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

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

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

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

Define $\mathcal{A}_\alpha$ as the event that there exists less than $\alpha l + 1$ successes in G trials. By employing the Chernoff Bound

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

Now, by using (6) and (7)

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

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

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

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

The methodology of Theorem 1 can be applied with (r, l)-*superstrings* of shorter length for stronger privacy guarantees. The following lemma provides a construction for the shortest (r, l)-*superstring* and evaluates its length.

Lemma 1: The length of the shortest (r, l)-*superstring* is equal to $r^l + l - 1$; that is, $f(r, l) = r^l + l - 1$.

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on December 15,2022 at 17:51:05 UTC from IEEE Xplore. Restrictions apply.

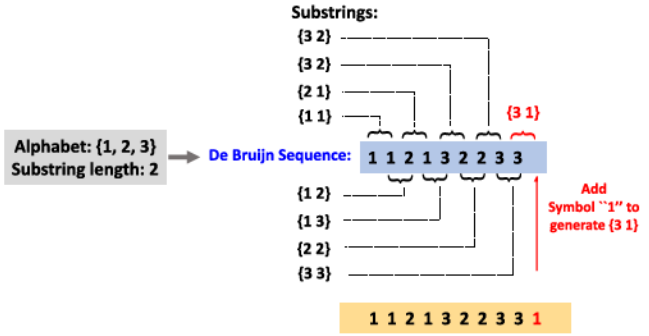

Fig. 4. Construction of a shortest $(3, 2)$-*superstring* by using a De Bruijn sequence $B(3, 2)$. The length of the constructed $(3, 2)$-*superstring* is $f(3, 2) = 3^2 + 2 - 1 = 10$.

Proof: We denote by $B(r, l)$ a De Bruijn sequence [79], [80] of order l on a size-r alphabet-$\mathcal{R}$. A De Bruijn sequence is a sequence with length r^l in which every possible length-l substring on $\mathcal{R}$ occurs exactly once as a contiguous subsequence, given that the last $(l-1)$ and the first $(l-1)$ letters of the De Bruijn sequence form a cyclic tail-to-head ligation for counting the substrings.

We construct a shortest (r, l)-*superstring* with length $(r^l + l - 1)$ from a chosen De Bruijn sequence $B(r, l)$ by repeating $B(r, l)$'s front $(l-1)$ symbols at the end of the sequence. 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 $B(r, l)$ in its first 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 noncyclic way, since the De Bruijn sequence's front $(l-1)$ symbols have been copied to its end (one example shown in Fig. 4). Thus, the constructed sequence contains all possible r^l substrings, and hence, by Definition 3, it is a valid (r, l)-*superstring*.

Next, we prove that the constructed sequence gives the shortest solution for an (r, l)-*superstring*. Each of the 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. ■

The solution for the shortest (r, l)-*superstring* is nonunique in general for $r \geq 2$ since we can construct our (r, l)-*superstring* by taking any De Bruijn sequence $B(r, l)$.

Shortest-Length Superstring-Based Obfuscation (SL-SBU): For each user we randomly (uniformly) choose a shortest-length *superstring* (as described above) and employ it for obfuscation. As noted earlier, if we reach the end of a *superstring*, another one is chosen uniformly at random.

Definition 5: Let $\mathcal{B}'_u$ be the event 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 2: The privacy performance when the shortest (r, l)-*superstring* is employed is given by

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

where

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

Proof: By using (5), we have

$$\mathbb{P}(\mathcal{B}'_u) \geq \mathbb{P}(\mathcal{E}'_u)\mathbb{P}(\mathcal{F}'_u) \tag{11}$$

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 note that the index values $L_{u,1}$ are equally likely over the first 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

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

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

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

In addition, similar to (9), $\mathbb{P}(\mathcal{F}'_u) = \mathbb{P}(\mathcal{F}_u)$, which, combined with (11) and (13), leads to (10). ■

Lemma 2: The lower bounds achieved by Theorems 1 and 2 are independent of the data sequence $\mathbf{X}$ if the data point set $\mathcal{R}$ is known.

Proof: This follows immediately from Theorems 1 and 2. ■

Theorems 1 and 2 provide ϵ-privacy for constant ϵ (i.e., ϵ not decreasing in the number of users n). As noted in Section III, this is a very strong version of privacy, and hence weaker forms are also of practical interest. Thus, we consider cases where ϵ goes to zero, but in a way that each user is still confused with $N^{(n)}$ users, where $N^{(n)} \to \infty$ as $n \to \infty$. First, the following lemma readily establishes that there are infinitely many users with the same pattern as user 1 in such cases.

Lemma 3: Let $N^{(n)}$ be the number of users with the same pattern as user 1. For any $0 < \beta < 1$, if $\mathbb{P}(\mathcal{B}'_u) = \mathbb{P}(\text{user } u \text{ has pattern of user 1}) \geq (1/n^{1-\beta})$, then $N^{(n)} \to \infty$ with high probability as $n \to \infty$. More specifically, as $n \to \infty$

$$\mathbb{P}\left(N^{(n)} \geq \frac{n^\beta}{2}\right) \to 1.$$

Proof: We define the binary random variable C_u to denote whether user u's obfuscation sequence $\mathbf{Z}_u$, for $u = 1, 2, \ldots, n$, contains user 1's identifying pattern. $C_u = 1$ indicates that Z_u contains user 1's identifying pattern, and $C_u = 0$ otherwise.

$N^{(n)}$ is the total number of users who have the same pattern as user 1's identifying pattern; thus $N^{(n)} = \sum_{u=1}^{n} C_u$. Recall

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on December 15,2022 at 17:51:05 UTC from IEEE Xplore. Restrictions apply.

that $\mathbb{P}(\mathcal{B}'_u) = \mathbb{P}(\text{user } u \text{ has pattern of user 1}) \geq (1/n^{1-\beta})$; hence

$$\mathbb{E}\left[N^{(n)}\right] = \Pi_{u=1}^{n}\mathbb{E}[C_u] \geq n\frac{1}{n^{1-\beta}} = n^{\beta}. \tag{14}$$

On the other hand, by employing the Chernoff bound, we have

$$\mathbb{P}\left(N^{(n)} \leq (1-\delta)\mathbb{E}\left[N^{(n)}\right]\right) \leq \exp\left(-\tfrac{\delta^2}{2}\mathbb{E}\left[N^{(n)}\right]\right). \tag{15}$$

Now if we assume $\delta = 0.5$, by (14) and (15), we can conclude

$$\mathbb{P}\left(N^{(n)} \leq \frac{n^{\beta}}{2}\right) \leq \mathbb{P}\left(N^{(n)} \leq \frac{\mathbb{E}\left[N^{(n)}\right]}{2}\right) \tag{16}$$

$$\leq \exp\left(-\frac{\mathbb{E}\left[N^{(n)}\right]}{8}\right) \tag{17}$$

$$\leq \exp\left(-\frac{n^{\beta}}{8}\right) \to 0 \tag{18}$$

as $n \to \infty$. As a result, as n becomes large, $\mathbb{P}(N^{(n)} \geq (n^{\beta}/2)) \to 1$. In other words, the total number of users who have the same pattern as user 1's identifying pattern goes to infinity. ■

The following theorem shows that by using the proposed SL-SBU technique, we can indeed achieve a privacy guarantee against pattern matching attacks while employing a small obfuscation probability, in fact with $p_{\text{obf}} \to 0$ as $n \to \infty$. As motivation, note that in many practical scenarios the size of the alphabet r could be very large and indeed can scale with n, the number of users. For example, consider a scenario where the data shows the location of users in an area of interest, such as a town or a neighborhood within a city. Assuming a certain level of granularity in the location data, the number of possible locations r and the number of users n become larger as the considered area becomes larger. In such scenarios, it makes sense to write $r = r(n)$ to explicitly denote that r can change as a function of n.

Below, Theorem 3 provides the solution for determining the obfuscation probability p_{obf} to achieve an infinite number of users who contain user 1's identifying pattern.

Theorem 3: For the SL-SBU method, let $l > 1$ and $h \geq 1$ be fixed. Choose $0 < \beta < 1$, and define $d(n) = m(n)n^{-(1-\beta/l-1)}$. If $[d(n)n^{\theta}]^{(1/l)} \leq r(n) \leq [d(n)n^{\theta l}]^{(1/l)}$ for some $0 < \theta < (1-\beta/l-1)$, then by choosing $p_{\text{obf}} = b_n = n^{-(1-\beta/l-1)+\theta}$, and $\liminf_{n\to\infty} mb_n \geq 9$, we have

$$\mathbb{P}(\mathcal{B}'_u) \geq \frac{c}{n^{1-\beta}}$$

for some constant $c = c(h, l)$.

Proof: First note that the assumptions $b_n = n^{-(1-\beta/l-1)+\theta}$, and $\liminf_{n\to\infty} mb_n \geq 9$ imply that $m(n) \to \infty$ as $n \to \infty$. Recall from Theorem 2 that

$$\mathbb{P}(\mathcal{B}'_u) \geq \frac{\left(1-(1-p_{\text{obf}})^h\right)^{(l-1)}}{r^l} \sum_{\alpha=0}^{\min\{(r^l-1),\lfloor Gp_{\text{obf}}\rfloor\}} 1-\exp\left(-\frac{\delta'^2_{\alpha}}{2}Gp_{\text{obf}}\right) \tag{19}$$

where

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

Note that for any $\tau \in \mathbb{R}$, $1-\tau \leq e^{-\tau}$; thus, with $\tau = b_n$

$$(1-b_n)^h \leq e^{-b_n h}, \quad \text{for } b_n \in \mathbb{R}. \tag{20}$$

In addition, for any $0 \leq \upsilon \leq 1$, $1 - e^{-\upsilon} \geq (\upsilon/2)$; thus, with $\upsilon = b_n h$

$$1 - e^{-b_n h} \geq \frac{b_n h}{2}, \quad \text{for } 0 \leq b_n h \leq 1. \tag{21}$$

Now, by (20) and (21), we can conclude

$$1-(1-b_n)^h \geq 1 - e^{-b_n h} \geq \frac{b_n h}{2},\ \text{for } 0 \leq b_n h \leq 1.$$

Note that h is a constant, and $b_n \to 0$ as $n \to \infty$. As a result

$$\frac{\left(1-(1-b_n)^h\right)^{l-1}}{r^l} \geq \left(\frac{b_n h}{2}\right)^{l-1} \cdot \frac{1}{r^l} = \left(\frac{h}{2}\right)^{l-1} \frac{b_n^{l-1}}{r^l}. \tag{22}$$

From the statement of the theorem, $[d(n)n^{\theta}]^{(1/l)} \leq r(n) \leq [d(n)n^{\theta l}]^{(1/l)}$ for some $0 < \theta < (1-\beta/l-1)$; thus

$$r^l \geq d(n)n^{\theta} = mn^{-\frac{1-\beta}{l-1}+\theta} = mb_n$$

as a result

$$r^l - 1 \geq mb_n - 1. \tag{23}$$

Since $G = m - h(l-1) \leq m$, we have

$$r^l - 1 \geq Gb_n - 1. \tag{24}$$

Also

$$Gb_n - 1 \leq \lfloor Gp_{\text{obf}}\rfloor = \lfloor Gb_n\rfloor. \tag{25}$$

Thus, by (24) and (25)

$$Gb_n - 1 \leq \min\left\{\left(r^l - 1\right), \lfloor Gp_{\text{obf}}\rfloor\right\} \leq \lfloor Gb_n\rfloor.$$

The above equation can be used to obtain a lower bound for the second term on the right side of (19)

$$\begin{aligned}
&\sum_{\alpha=0}^{\min\{(r^l-1),\lfloor Gp_{\text{obf}}\rfloor\}} \left\{1-\exp\left(-\frac{1}{2}\delta'^2_{\alpha}Gp_{\text{obf}}\right)\right\} \\
&= \min\left\{\left(r^l-1\right), \lfloor Gp_{\text{obf}}\rfloor\right\} + 1 \\
&\quad - \sum_{\alpha=0}^{\min\{r^l-1,\lfloor Gp_{\text{obf}}\rfloor\}} \exp\left(-\frac{1}{2}\delta'^2_{\alpha}Gp_{\text{obf}}\right) \\
&\geq Gb_n - \sum_{\alpha=0}^{\lfloor Gb_n\rfloor} \exp\left(-\frac{1}{2}\delta'^2_{\alpha}Gb_n\right).
\end{aligned} \tag{26}$$

Now since $G = m - h(l-1)$, h and l are constants, and $b_n \to 0$ as $n \to \infty$

$$\begin{aligned}
\liminf_{n\to\infty} Gb_n &= \liminf_{n\to\infty}(mb_n - h(l-1)b_n) \\
&= \liminf_{n\to\infty} mb_n \geq 9.
\end{aligned}$$

Thus, for large enough n, $Gb_n \geq 8$. Note that for $\alpha = 0, \ldots, \lfloor (Gb_n/2)\rfloor$

$$\delta'_{\alpha} = 1 - \frac{\alpha}{Gb_n} \geq 1 - \frac{1}{2} = \frac{1}{2}$$

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on December 15,2022 at 17:51:05 UTC from IEEE Xplore. Restrictions apply.

thus, for $\alpha = 0, \ldots, \lfloor (Gb_n/2) \rfloor$

$$\exp\left(-\frac{1}{2}\delta_\alpha'^2 Gb_n\right) \le \exp\left(-\frac{Gb_n}{8}\right) \le \exp(-1). \quad (27)$$

On the other hand, for $\alpha = \lfloor (Gb_n/2) \rfloor + 1, \ldots, \lfloor Gb_n \rfloor$

$$\delta'_\alpha = 1 - \frac{\alpha}{Gb_n} \ge 0$$

and as a result, for $\alpha = \lfloor (Gb_n/2) \rfloor + 1, \ldots, \lfloor Gb_n \rfloor$

$$\exp\left(-\frac{1}{2}\delta_\alpha'^2 Gb_n\right) \le 1. \quad (28)$$

Now by (27) and (28), we conclude:

$$\sum_{\alpha=0}^{\lfloor Gb_n \rfloor} \exp\left(-\frac{1}{2}\delta_\alpha'^2 Gb_n\right) \le \left(\frac{Gb_n}{2}+1\right)\exp(-1) + \left(\frac{Gb_n}{2}+1\right) \times 1 \quad (29)$$

$$= \frac{Gb_n}{2}\left(1 + \exp(-1) + \frac{2(1+\exp(-1))}{Gb_n}\right) \quad (30)$$

$$\le \frac{Gb_n}{2}\left(1 + \exp(-1) + \frac{2(1+\exp(-1))}{8}\right) \quad (31)$$

$$\le 0.86 Gb_n. \quad (32)$$

As a result, by (26) and (32), for large enough n

$$\sum_{\alpha=0}^{\min\{(r^l-1), \lfloor Gp_{\text{obf}} \rfloor\}} \left\{1 - \exp\left(-\frac{1}{2}\delta_\alpha'^2 Gp_{\text{obf}}\right)\right\} \ge 0.14 Gb_n \ge 0.1 Gb_n.$$

Since $G = m - h(l-1) \le m$, where h and l are constants, and $m \to \infty$, we conclude for large enough n, $G(n) \ge (m(n)/2)$

$$\sum_{\alpha=0}^{\min\{(r^l-1), \lfloor Gp_{\text{obf}} \rfloor\}} \left\{1 - \exp\left(-\frac{1}{2}\delta_\alpha'^2 Gp_{\text{obf}}\right)\right\} \ge 0.05 mb_n. \quad (33)$$

Now, by (19), (22), and (33), we conclude that for some constant $c = c(h, l)$

$$\mathbb{P}(\mathcal{B}'_u) \ge \left(\frac{h}{2}\right)^{l-1} \frac{b_n^{l-1}}{r(n)^l} \frac{Gb_n}{10} \ge c\frac{mb_n^l}{r(n)^l}.$$

Since $r(n)^l \le d(n)n^{\theta l} = mn^{-(1-\beta/l-1)} \times n^{\theta l}$, and $b_n = n^{-(1-\beta/l-1)+\theta}$

$$\mathbb{P}(\mathcal{B}'_u) \ge cmb_n^l \times \frac{n^{\frac{1-\beta}{l-1}}}{m \times n^{\theta l}} = c\frac{n^{\frac{1-\beta}{l-1}}}{n^{\frac{1-\beta}{l-1}l-\theta l}n^{\theta l}} = \frac{c}{n^{\frac{1-\beta}{l-1}(l-1)}} = \frac{c}{n^{1-\beta}}.$$

■

The SL-SBU and i.i.d. obfuscation schemes will be compared extensively via simulation in Section VIII. Here, we provide an analytical result to both predict the results of that comparison and provide insight into such.

Theorem 4: Suppose the sequence is $[X_1, X_2, \ldots]$ and the pattern is $Q = [q_1, q_2, \ldots, q_l]$. We say the pattern occurs[1] in the sequence at time index t if $X_t = q_1$, $X_{t+1} = q_2$, …, $X_{t+l-1} = q_l$. We use $T_{\text{SL-SBU}}$ to denote the time index where the pattern first occurs in the SL-SBU obfuscation sequence. Similarly, we use $T_{\text{i.i.d.}}$ to denote the time index where the pattern first occurs in the i.i.d. obfuscation sequence. Then, the expectation of these times are given by

$$\mathbb{E}[T_{\text{SL-SBU}}] = \frac{r^l+1}{2}$$
$$\mathbb{E}[T_{\text{i.i.d.}}] \ge r^l.$$

Proof: For the SL-SBU obfuscation sequence with length $r^l + l - 1$, we can immediately establish the result by the property of the De Bruijn-based sequences of Theorem 2

$$\mathbb{E}[T_{\text{SL-SBU}}] = \frac{1}{r^l}\left[1 + 2 + \cdots + r^l\right] = \frac{r^l+1}{2}.$$

Now, we consider the i.i.d. obfuscation sequence, and the corresponding $\mathbb{E}[T_{\text{i.i.d.}}]$. We say that the pattern Q has an overlap of length $l' < l$ if

$$q_1 = q_{l-l'+1}, q_2 = q_{l-l'+2}, q_{l'} = q_l.$$

The largest such l' is called the overlap value of the sequence Q and shown by l_Q; thus, for every pattern of length l, we have $0 \le l_Q \le l-1$. Now the arrival times of pattern Q in the i.i.d. sequence $[X_1, X_2, \ldots]$ can be modeled as a delayed renewal process with an average interarrival time μ. By Blackwell's theorem for delayed renewal processes [81], we have

$$\lim_{t\to\infty} P(\text{Renewal at } t) = \frac{1}{\mu}.$$

Since the sequence is i.i.d., we also have

$$\lim_{t\to\infty} P(\text{Renewal at } t) = \frac{1}{r^l}.$$

We conclude $\mu = r^l$. Now, consider two cases: $l_Q = 0$ and $l_Q > 0$. If $l_Q = 0$, then

$$\mathbb{E}[T_{\text{i.i.d.}}] = \mu = r^l.$$

On the other hand, if $l_Q > 0$, let $Q' = [q_1, q_2, \ldots, q_{l_Q}]$. In this case, let al.o $T_{\text{i.i.d.}}(Q')$ denote the time index where the pattern Q' first occurs in the i.i.d. obfuscation sequence. We have

$$\mathbb{E}[T_{\text{i.i.d.}}] = \mathbb{E}\left[T_{\text{i.i.d.}}(Q')\right] + \mu \ge \mu = r^l.$$

■

Finally, we note that the i.i.d. obfuscation approach of [22] can be readily combined with the techniques proposed here to provide robust privacy simultaneously against both statistical matching and pattern matching attacks.

V. Combination of I.I.D. Obfuscation and SL-SBU Obfuscation

In this section, we introduce the concept of *perfect privacy* from [22, Th. 2], which provides a strong privacy guarantee if a statistical model for the data is known. After combining *perfect privacy* and our proposed obfuscation method, user 1 is able to achieve privacy against pattern matching attacks and

[1]Here, "the pattern occurs" is the specific case of Definition 1 when $h = 1$.

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on December 15,2022 at 17:51:05 UTC from IEEE Xplore. Restrictions apply.

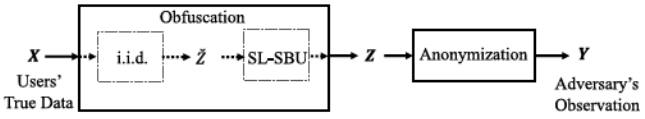

Fig. 5. Applying two stages of obfuscation and then anonymization to the users' data points.

perfect privacy with the total cost given by a modest noise level provided in Theorem 5.

To this point, we have employed the SL-SBU obfuscation method to protect users' data against pattern matching attacks while the adversary makes no assumptions about the statistical model of users' data sequences. However, this method has a drawback: as the number of possible values for each user's data points ($r(n)$) increases, it becomes exponentially less likely that an identifying pattern of a user is observed within other users' data; as a result, a pattern matching attack would become a serious threat to users' privacy.

On the other hand, Takbiri *et al.* [22] considered a strong assumption regarding the statistical model of users' data and introduced a simple i.i.d. obfuscation method in which the samples of the data of each user are reported with error with a certain probability, where that probability itself is generated randomly for each user. In other words, the obfuscated data is obtained by passing the users' data through an r-ary symmetric channel with a random error probability. Takbiri *et al.* [22] demonstrated that if the amount of noise level is greater than a critical value, users have perfect privacy against all of the adversary's possible attacks. The definition of perfect privacy is adopted from [16].

Definition 6: User u has *perfect privacy* at time k if and only if

$$\lim_{n\to\infty} \mathbb{I}(X_u(k); \mathbf{Y}) = 0$$

where $\mathbb{I}(X_u(k); \mathbf{Y})$ denotes the mutual information between the data point of user u at time k and the collection of the adversary's observations for all of the users.

Here, we will combine these two methods of obfuscation in order to benefit from the advantages of both methods and achieve perfect privacy. Note that combining these two techniques does not have any cost asymptotically.

As shown in Fig. 5, two stages of obfuscations and one stage of anonymization are employed to achieve perfect privacy. Note that the first stage is the same i.i.d. obfuscation technique given in [22, Th. 2], and the second stage of obfuscation is the SL-SBU method introduced previously. Thus, in Fig. 5, $\check{Z}_u(k)$ shows the (reported) data point of user u at time k after applying the first stage of obfuscation with the noise level equal to $a_n = \Omega(n^{-(1/r-1)})$, and $Z_u(k)$ shows the (reported) data point of user u at time k after applying the second stage of obfuscation with the noise level equal to $b_n = \Omega(n^{-(1-\beta/l-1)})$. Define the noise level of a two-stage obfuscation scheme with independent obfuscation probabilities a_n and b_n as $\psi_n = a_n + b_n - a_n b_n$. We then have the following result.

Theorem 5: If $\mathbf{Z}$ is the obfuscated version of $\mathbf{X}$ after two stages of obfuscation, and $\mathbf{Y}$ is the anonymized version of $\mathbf{Z}$, and

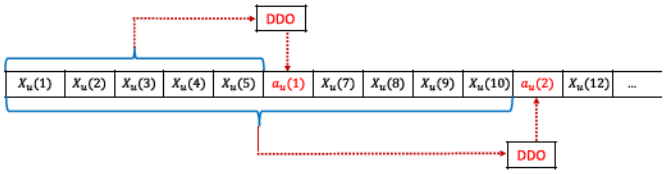

Fig. 6. DDO: $a_u(j)$ is chosen by the DDO algorithm based on the realized obfuscated sequence so far.

1) the length of the time series data, $m = m(n)$, is arbitrary;
2) the noise level of the obfuscation method is $\psi_n = \Omega(\max\{n^{-(1/r-1)}, n^{-(1-\beta/l-1)}\})$.

Then, user 1 has:

1) *privacy against pattern matching attacks* in any case;
2) *perfect privacy* if the assumptions about the statistical model of users' data is accurate.

Proof: First, we show that if the assumptions for the statistical model of users' data is accurate, users will have perfect privacy. We employ a noise level for the first stage of obfuscation equal to $a_n = \Omega(n^{-(1/r-1)})$, and the noise level for the second stage of obfuscation equal to $b_n = \Omega(n^{-(1-\beta/l-1)})$. Using the definition for the noise level for two-stage obfuscation given before the theorem statement, the noise level of the combined obfuscation mechanism is

$$\psi_n = \Omega(\max\{a_n, b_n\}) = \Omega\left(\max\left\{n^{-\frac{1}{r-1}}, n^{-\frac{1-\beta}{l-1}}\right\}\right) \quad (34)$$

as $n \to \infty$. From [22, Th. 2]: if a_n is significantly larger than $(1/n^{r-1})$, then all users have perfect privacy independent of the value of $m(n)$. Now, since $\psi_n \geq a_n$, by employing a noise value equal to $\psi_n = \Omega(\max\{n^{-(1/r-1)}, n^{-(1-\beta/l-1)}\})$, all users achieve perfect privacy independent of the value of $m(n)$. In other words, as $n \to \infty$, $\mathbb{I}(X_u(k); \mathbf{Y}) = 0$.

If the assumption regarding the statistical model of users' data is accurate or not, Theorem 3 establishes that users would have privacy against pattern matching attack due to the second stage of our obfuscation method. ■

VI. Data-Dependent Obfuscation

The obfuscation techniques proposed in Section IV are independent of the user data, as would be appropriate for real-time operation on nonbuffered data as discussed in Section I. However, as also discussed in Section I, there are scenarios such as image processing where the entire data sequence might be known to the PPM. To exploit such, we employ opportunistic *superstring* creation, which we refer to as DDO.[2] The key point here is to choose obfuscated values $a_u(j)$ in an opportunistic fashion; that is, at each point, the element $a_u(j)$ in the *superstring* is chosen based on the realized obfuscated sequence so far, with the goal of choosing $a_u(j)$ in a way to maximize the number of distinct patterns in the obfuscated sequence of user u. Fig. 6 shows the structure of the DDO algorithm.

[2]Note that the sequences developed might not technically be *superstrings*, as defined formally in Section IV, but, since the sequences are employed in a similar fashion to the *superstrings* of Section IV, we employ the same term to avoid confusion.

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on December 15,2022 at 17:51:05 UTC from IEEE Xplore. Restrictions apply.

A. DDO Algorithms to Thwart Pattern Matching Attacks

First, we formulate solutions in a general setting for arbitrary pattern length l. Then, we design three examples of DDO algorithms for specific values of l. Indeed, some practical pattern matching attacks are based on patterns with a small length for identification or classification [82], [83], and, as validated in our simulation results later, low-order DDOs can also work well for larger l than they were designed.

In what follows we drop the subscript u to simplify notation. Hence, let $X(k)$ be the data point at index k for an arbitrary user and $Z(k)$ be its obfuscated version.

Definition 7: For a *pattern* Q of length l, $N_k^l(Q)$ is the total number of times the *pattern* Q has been observed as a *pattern* (Definition 1) up to and including time k. That is, it is the number of times the *pattern* has been observed in

$$Z(1), Z(2), \ldots, Z(k).$$

A pattern distribution N_k^l for the sequence $Z(1), Z(2), \ldots, Z(k)$ is the collection of values $N_k^l(Q)$ across all *patterns* Q of length l.

In the special case of $l = 1$, $N_k^1(i)$ is the number of times the value i has appeared in $Z(1), Z(2), \ldots, Z(k)$.

Definition 8: A DDO algorithm of order l is a mapping from the set of all pattern distributions to the set of probability distributions over data point set $\mathcal{R} = \{0, 1, \ldots, r-1\}$. This probability distribution, denoted by P_{DDO}, provides the probability of obfuscating $X(k+1)$ to the values $0, 1, \ldots, r-1$, given that we are performing obfuscation on a given data sample.

The simplest DDO algorithm, which we call least-observed value (LOV), works as follows: to obfuscate $X(k+1)$, choose one of the values not present in $Z(1), Z(2), \ldots, Z(k)$, uniformly at random. If all values have been observed, we obfuscate the data points with a value drawn uniformly at random from $\mathcal{R}$. To execute the algorithm, we only need to keep the subset of $\mathcal{R}$ containing the values that have not been observed to this point and choose one of them at random for obfuscation. Denote $\mathbb{P}_{\text{LOV}}$ as the probability that the obfuscated sequence has user 1's identifying pattern after applying the LOV algorithm. For $l = 1$

$$\mathbb{P}_{\text{LOV}} \geq \sum_{k=0}^{r-1} \binom{m}{k} p_{\text{obf}}^k (1 - p_{\text{obf}})^{m-k} \left(\frac{k}{r}\right) + \sum_{k=r}^{m} \binom{m}{k} p_{\text{obf}}^k (1 - p_{\text{obf}})^{m-k}$$

where m is the length of the sequence.

The second DDO algorithm, which we term probabilistic LOV (PLOV), is in some sense a generalization of the LOV algorithm that introduces more randomness in the operation. The intuition behind the obfuscation of PLOV is to give a higher probability to the values that have appeared less so far. Specifically, at time k, define

$$\tilde{q}_i = \left(\frac{N_k^1(i)}{k}\right)^{\gamma}$$

where $0 < \gamma$ is a design parameter. A typical value is $\gamma = (1/10)$. Now let

$$q_i = \frac{\tilde{q}_i}{\sum_{j=1}^{r} \tilde{q}_j}$$

TABLE I
TIME COMPLEXITY AND SPACE COMPLEXITY OF EACH OBFUSCATION METHOD FOR EACH USER'S DATA SEQUENCE WITH LENGTH m

DIO algorithms	i.i.d.	SL-SBU
time complexity	$O(m)$	$O(m)$
space complexity	$O(r)$	$O(r^l)$

DDO algorithms	LOV	PLOV	MANP
time complexity	$O(m \log r)$	$O(mr)$	$O(m \cdot \max\{r, h\} \cdot \log r)$
space complexity	$O(r)$	$O(r)$	$O(r \cdot \max\{r, h\})$

$$q_{\max} = \max\{q_i, i = 0, 1, \ldots, r-1\}$$
$$q_{\min} = \min\{q_i, i = 0, 1, \ldots, r-1\}$$

and choose

$$b \leq \min\left(\frac{1}{rq_{\max} - 1}, \frac{r-1}{1 - rq_{\min}}\right).$$

For example, we set $b = 0.99 \min((1/[rq_{\max} - 1]), (r - 1/[1 - rq_{\min}]))$ in our experiments. The obfuscation probabilities are given by

$$p_i = \frac{1+b}{r} - bq_i, \quad i = 0, 1, 2, \ldots, r-1$$

where p_i is the conditional probability of obfuscating to i ($Z_u(k+1) = i$), given that we are obfuscating $X_u(k+1)$.

The third DDO algorithm is termed make a new pattern (MANP), which chooses a value that completes as many patterns as possible with length l that have not been observed to this point. Specifically, we choose the value $l = 2$.

Define $\mathfrak{P}_k$ as the total number of distinct patterns of length $l = 2$ observed in the obfuscated sequence until time k [i.e., in $Z(1), Z(2), \ldots, Z(k)$]. Thus, $\mathfrak{P}_1 = 0$ and $\mathfrak{P}_2 = 1$. Also, for $i \in \mathcal{R}$, define $\mathfrak{P}_{k+1}(i)$ as the value of $\mathfrak{P}_{k+1}$ given that $Z(k+1) = i$. Given we are obfuscating at time $k+1$, choose

$$Z(k+1) = \arg\max \mathfrak{P}_{k+1}(i), \quad i \in \mathcal{R}.$$

These three DDO algorithms (LOV, PLOV, MANP) will be simulated in the next section.

VII. COMPLEXITY ANALYSIS

The time complexity and space complexity for each obfuscation algorithm (DIO and DDO) for each user's data sequence with length m are shown in Table I based on the following assumptions.

1) We assume searching/insertion time complexity for a specific element in/to a set with size N is $O(\log N)$.
2) We assume the sorting algorithm takes $O(N \log N)$ time complexity for an array with size N.

The time complexity for each of the two DIO algorithms is $O(m)$, since each obfuscates each data point based on the obfuscation sequence, which takes constant time. For LOV, it takes $O(\log r)$ time to check (search) if the current obfuscated data point is (or is not) a member of the letter set

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on December 15,2022 at 17:51:05 UTC from IEEE Xplore. Restrictions apply.

TABLE II
NUMERICAL EVALUATION OF THE LOWER BOUNDS OF THEOREM 1 (ϵ) AND THEOREM 2 (ϵ') FOR THE PERCENTAGE OF SEQUENCES THAT CONTAIN A USER'S IDENTIFYING PATTERN WHEN THE PROPOSED SBU OBFUSCATION APPROACHES ARE EMPLOYED

m	r	l	h	p_{obf}	lower bound ϵ	lower bound ϵ'
1000	20	3	10	10%	0.15%	0.45%
1000	20	3	8	10%	0.12%	0.35%
1000	20	3	10	15%	0.36%	1.06%
1000	20	3	10	30%	1.07%	3.22%
4000	20	3	10	10%	0.66%	1.98%
10000	20	3	10	10%	1.69%	5.08%
1000	20	2	10	10%	7.12%	14.17%
1000	20	2	8	10%	6.24%	12.41%
1000	20	2	10	15%	13.47%	26.84%
1000	20	2	10	30%	33.57%	67.02%
2000	20	2	10	10%	14.84%	29.60%
4000	20	2	10	10%	30.52%	60.97%

which have been seen before (with worst-case size r). For MANP, it needs to search in the set of patterns which have been seen before [with worst-case size $O(r^2)$ and search time $O(\log r^2) = O(\log r)$] for each candidate letter (with r different choices in the data point set $\mathcal{R}$, $O(r \log r)$ in total). And, it also takes $O(r \log r)$ for sorting the candidate letters by the order of their achievable patterns if used for obfuscating at the current data point. Finally, it takes $O(h \log r^2) = O(h \log r)$ time complexity for inserting the new pattern list into the set of patterns previously observed after obfuscation (worst-case size h). For PLOV, it takes $O(r)$ complexity for each obfuscation operation due to the summation of the elements of vector $\tilde{q}$ with size r. For space complexity, the i.i.d., LOV and PLOV methods each take $O(r)$ space for storing the data point set. For the SL-SBU method, it takes $O(r^l)$ space to store the obfuscation sequence (De Bruijn sequence) with length r^l (the number of all possible patterns); for MANP, it takes $O(r^2)$ space for storing the set of patterns which have been seen before (with worst-case size r^2) and $O(hr)$ space for storing the counting table for each letter's contribution to create new patterns (maximum size for each letter is equal to h, $O(hr)$ in total).

VIII. NUMERICAL RESULTS AND VALIDATION

We evaluate the performance of the proposed SBU methods and the three DDO algorithms on synthetic i.i.d. data sequences and on sequences from the Reality Mining data set. The data points in the i.i.d. data sequence for each user are drawn independently and identically from the data point set $\mathcal{R}$. Reality Mining is a data set released by the MIT Media Laboratory which tracks a group of 106 (anonymized) mobile phone users [84]. The Reality Mining data set contains traces of users' associated cell tower IDs across time. Here, we further sample the data traces with sampling interval at least 10 min to avoid significant data point repetition.

TABLE III
SIMULATION RESULTS FOR THE CASE OF I.I.D. DATA SEQUENCES DRAWN FROM AN ALPHABET OF SIZE r, WHEN USING AN I.I.D. SEQUENCE AND THE SL-SBU SEQUENCE FOR OBFUSCATION: THE FRACTION OF SEQUENCES WHICH CONTAIN USER 1'S IDENTIFYING PATTERN ($[r-l+1, \ldots, r-1, r]$) FOR $h = 10$ AND $p_{\text{OBF}} = 10\%$

m	r	l	h	p_{obf}	fraction (i.i.d.)	fraction (SL-SBU)
10^3	20	2	10	10%	0.2185	0.7380
10^4	20	2	10	10%	0.9097	1
10^4	20	3	10	10%	0.1176	0.2571
10^5	20	3	10	10%	0.6949	0.9598
10^3	30	2	10	10%	0.1091	0.5853
10^4	30	2	10	10%	0.6624	0.9999
10^5	30	3	10	10%	0.3042	0.7656
10^6	30	3	10	10%	0.9712	1
10^3	40	2	10	10%	0.0666	0.4838
10^4	40	2	10	10%	0.4621	0.9983
10^5	40	3	10	10%	0.1465	0.6010
10^6	40	3	10	10%	0.7838	0.9999
10^3	50	2	10	10%	0.0462	0.4142
10^4	50	2	10	10%	0.3301	0.9913
10^5	50	3	10	10%	0.0808	0.4937
10^6	50	3	10	10%	0.5412	0.9994

A. Evaluation for SBU Obfuscation

We consider the numerical evaluation of the achievable lower bounds, as given in Theorems 1 and 2 (SL-SBU, optimized version), for the fraction of sequences that contain a potentially identifying pattern of user 1 when using the proposed SBU obfuscation approach. We use ϵ and ϵ' to denote these two lower bounds, respectively, and the results are shown in Table II. Note that these are deterministic numerical evaluations of the bounds in Theorems 1 and 2. Hence, the results show that the proposed PPMs will result in a nonzero percentage of the user set $\mathcal{U}$ that have any potentially identifying pattern of user 1 in their obfuscated sequences with high probability, and we can observe that the SL-SBU obfuscation sequence has a higher lower bound than the regular SBU obfuscation sequence. As expected, increasing the data sequence length m or the obfuscation noise level p_{obf} will increase the chance of observing the pattern in the obfuscated sequences for both methods. The advantage of the SL-SBU approach over the regular SBU approach becomes more significant as longer sequences are considered.

Next, we test the effectiveness of the i.i.d. obfuscation and SL-SBU obfuscation approaches on synthetic i.i.d. sequences. We believe the i.i.d. sequences yield the worst-case scenario: there is no dependency between any two consecutive data points that would lead to common subsequences of potentially identifying patterns being likely to be shared across

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on December 15,2022 at 17:51:05 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
SIMULATION RESULTS FOR THE CASE OF I.I.D. DATA SEQUENCES DRAWN FROM AN ALPHABET OF SIZE r, WHEN USING AN I.I.D. SEQUENCE AND THE SL-SBU SEQUENCE FOR OBFUSCATION: THE FRACTION OF SEQUENCES WHICH CONTAIN USER 1'S IDENTIFYING PATTERN ($[r-l+1,\ldots,r-1,r]$) FOR $h=5$ AND $p_{\text{OBF}}=10\%$

m	r	l	h	p_{obf}	fraction (i.i.d.)	fraction (SL-SBU)
10^3	20	2	5	10%	0.1223	0.3733
10^4	20	2	5	10%	0.7078	0.9932
10^4	20	3	5	10%	0.0370	0.0391
10^5	20	3	5	10%	0.2683	0.2961
10^3	30	2	5	10%	0.0607	0.2585
10^4	30	2	5	10%	0.4268	0.9587
10^5	30	3	5	10%	0.0949	0.1194
10^6	30	3	5	10%	0.5891	0.7174
10^3	40	2	5	10%	0.0383	0.1976
10^4	40	2	5	10%	0.2719	0.8846
10^5	40	3	5	10%	0.0438	0.0646
10^6	40	3	5	10%	0.3271	0.4724
10^3	50	2	5	10%	0.0277	0.1616
10^4	50	2	5	10%	0.1868	0.8020
10^5	50	3	5	10%	0.0268	0.0429
10^6	50	3	5	10%	0.1840	0.3170

TABLE V
SIMULATION RESULTS FOR THE CASE OF I.I.D. DATA SEQUENCES DRAWN FROM AN ALPHABET OF SIZE r, WHEN USING AN I.I.D. SEQUENCE AND THE SL-SBU SEQUENCE FOR OBFUSCATION: THE FRACTION OF SEQUENCES WHICH CONTAIN USER 1'S IDENTIFYING PATTERN ($[r-l+1,\ldots,r-1,r]$) FOR $h=10$, $p_{\text{OBF}}=5\%$

m	r	l	h	p_{obf}	fraction (i.i.d.)	fraction (SL-SBU)
10^3	20	2	10	5%	0.0673	0.2203
10^4	20	2	10	5%	0.4622	0.9255
10^4	20	3	10	5%	0.0235	0.0259
10^5	20	3	10	5%	0.1502	0.1758
10^3	30	2	10	5%	0.0358	0.1497
10^4	30	2	10	5%	0.2454	0.7885
10^5	30	3	10	5%	0.0539	0.0758
10^6	30	3	10	5%	0.3693	0.5194
10^3	40	2	10	5%	0.0241	0.1147
10^4	40	2	10	5%	0.1509	0.6639
10^5	40	3	10	5%	0.0274	0.0444
10^6	40	3	10	5%	0.1770	0.3148
10^3	50	2	10	5%	0.0187	0.0930
10^4	50	2	10	5%	0.1025	0.5736
10^5	50	3	10	5%	0.0184	0.0314
10^6	50	3	10	5%	0.1015	0.2150

users. Furthermore, to consider (pessimistically) only patterns that are inserted via our obfuscation method (eliminating the possibility that a user trace already has the desired pattern), we make certain that a data set $\mathcal{R}$ with size r has a unique sequence by assigning user 1 a unique pattern and drawing other users' sequences from the subset of the data set with size $(r-l)$; for instance, if the pattern length is $l=3$, we insert a pattern $[r-2, r-1, r]$ into user 1's sequence at a random place for uniqueness. We then follow the obfuscation procedure from Section III for each iteration and calculate the results by averaging the fraction of sequences which contain user 1's identifying pattern (by Definition 1) for all iterations. The validation results for different parameter settings are shown in Tables III, IV, and V. From the overall results, we can observe that the SL-SBU obfuscation sequence performs better than the i.i.d. obfuscation sequence, as predicted by Theorem 4.

B. Evaluation of the Data-Dependent and Data-Independent Obfuscation Algorithms

Next, we consider the simulation of the proposed DIO obfuscation methods (SL-SBU and i.i.d. obfuscation sequences) and the three DDO algorithms (LOV, PLOV, and MANP) on i.i.d. data sequences and the Reality Mining data set. Recall the three DDO obfuscation algorithms' design: the LOV algorithm chooses the obfuscation value which has not been observed in the user's obfuscated sequence before; the PLOV algorithm selects obfuscating values that have been less observed in the user's obfuscated sequence with higher probability; MANP chooses the obfuscating letter which completes the most previously unobserved patterns with length $l=2$.

Figs. 7 and 8 show the performance comparison of the obfuscation algorithms on i.i.d. sequences and the Reality Mining data set, respectively, for the pattern length $l=1$. In this case, the LOV algorithm has the best performance on both i.i.d. sequences and data set sequences, although all three DDO algorithms achieve very good performance with limited p_{obf}. For the two DIO algorithms, their performance is not affected by the data sequence's type since they are data-independent algorithms. For the DDO algorithms, their performances are more affected by the types of the data sequence when p_{obf} is relatively small. For instance, when p_{obf} is between 0.004 and 0.02, the DDO algorithms' performances would not be boosted and stable enough until p_{obf} is large enough (greater than 0.02). The reason behind this is that for the data set sequence, the letters appearing in the previous obfuscated data points are not as various as in the i.i.d. sequence data. Thus, their performances are degraded on the data set sequences. For the two DIO algorithms, the SL-SBU obfuscation's performance is better than the i.i.d. obfuscation's performance for both i.i.d. sequences and the Reality Mining data set since SL-SBU employs the optimal obfuscation sequence with shortest length (De Bruijn sequence): the SL-SBU obfuscation method can achieve a fraction of nearly 0.90 with $p_{\text{obf}}=0.02$, while

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on December 15,2022 at 17:51:05 UTC from IEEE Xplore. Restrictions apply.

TABLE VI
NUMERICAL RESULTS OF THE SL-SBU OBFUSCATION IN COMPARISON WITH THE EXISTING PRIVACY-PROTECTING MECHANISMS—GENERALIZATION AND SUBSAMPLING—ON THE REALITY MINING DATA SET: THE AVERAGING FRACTION OF SEQUENCES WHICH CONTAIN EACH USER'S IDENTIFYING PATTERN. $r = 20$, $m = 1000$, $l = 2$, $h = 10$. (a) SL-SBU OBFUSCATION. (b) GENERALIZATION METHOD. (c) SUBSAMPLING METHOD

(a)

p_{obf}	0.0	0.1	0.2	0.3	0.4	0.5
fraction	0.1490	0.5062	0.7202	0.8811	0.9921	0.9998

(b)

group size	1	2	3	4	5
fraction	0.1429	0.4580	0.4974	0.7511	0.7674

(c)

T_s	10 mins	12 mins	14 mins	16 mins	18 mins	20 mins	25 mins	30 mins
fraction	0.1442	0.1435	0.1429	0.1471	0.1524	0.1530	0.1604	0.1668

TABLE VII
NUMERICAL RESULTS OF THE SL-SBU OBFUSCATION IN COMPARISON WITH THE EXISTING PRIVACY-PROTECTING MECHANISMS—GENERALIZATION AND SUBSAMPLING—ON THE REALITY MINING DATA SET: THE AVERAGING FRACTION OF SEQUENCES WHICH CONTAIN EACH USER'S IDENTIFYING PATTERN. $r = 20$, $m = 1000$, $l = 3$, $h = 10$. (a) SL-SBU OBFUSCATION. (b) GENERALIZATION METHOD. (c) SUBSAMPLING METHOD

(a)

p_{obf}	0.0	0.1	0.2	0.3	0.4	0.5
fraction	0.0449	0.1796	0.2812	0.3555	0.4135	0.4780

(b)

group size	1	2	3	4	5
fraction	0.0456	0.2760	0.3237	0.6253	0.6568

(c)

T_s	10 mins	12 mins	14 mins	16 mins	18 mins	20 mins	25 mins	30 mins
fraction	0.0444	0.0455	0.0436	0.0475	0.0507	0.0498	0.0504	0.0537

the i.i.d. obfuscation method achieves a fraction around 0.60 fraction with $p_{\text{obf}} = 0.02$.

Fig. 9 shows the performance comparison of the obfuscation algorithms on i.i.d. sequences with pattern length $l = 2$. In this case, for the DDO algorithms, the MANP algorithm has the best performance, then the PLOV algorithm, while the LOV algorithm has the poorest performance among them since it does not intend to create patterns with length $l = 2$. For the two DIO algorithms, the SL-SBU obfuscation's performance is better than that of the i.i.d. obfuscation's for both the i.i.d. sequences and the Reality Mining data set. For the Reality Mining data set validation results, as shown in Fig. 10, MANP's performance is poorer on the realistic data set compared to the i.i.d. sequence due to the sparseness and repetition of the data points in the real data sequences. For the two DIO algorithms, their performance is again not affected by the data sequence's type. The SL-SBU obfuscation's performance is better than the i.i.d. obfuscation's for both of the i.i.d. sequences and the Reality Mining data set since the SL-SBU obfuscation sequence intends to create patterns in an optimal way. For instance, the SL-SBU obfuscation method can achieve a fraction of nearly 0.70 with $p_{\text{obf}} = 0.10$, while the i.i.d. obfuscation method achieves a fraction around 0.20 with the same $p_{\text{obf}} = 0.10$.

Fig. 11(a) and (b) show performances of the obfuscation algorithms on the i.i.d. sequences and the Reality Mining data set, respectively, when the pattern length $l = 3$. Among all three DDO algorithms, the PLOV has the best performance on i.i.d. sequences since it is the most robust to pattern length; the PLOV and the MANP have similar performance on data set sequences. For the DIO algorithms, the SL-SBU obfuscation has the best performance over all other methods either on the i.i.d. sequences (for large enough p_{obf}) or the Reality Mining sequences, due to its robustness across different data sequence types, which proves that the SL-SBU obfuscation is able to deterministically create all patterns without depending on the obfuscated data points and will eventually create any pattern if the sequence length is long enough or the obfuscation probability p_{obf} is high enough. In contrast, none of the DDO algorithms are designed specifically for pattern length above 2, so their performances would be limited. We can also observe that all three DDO algorithms have poorer performances on the Reality Mining data set than the i.i.d. sequences since the variation of the data points in the data set is limited and hence

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on December 15,2022 at 17:51:05 UTC from IEEE Xplore. Restrictions apply.

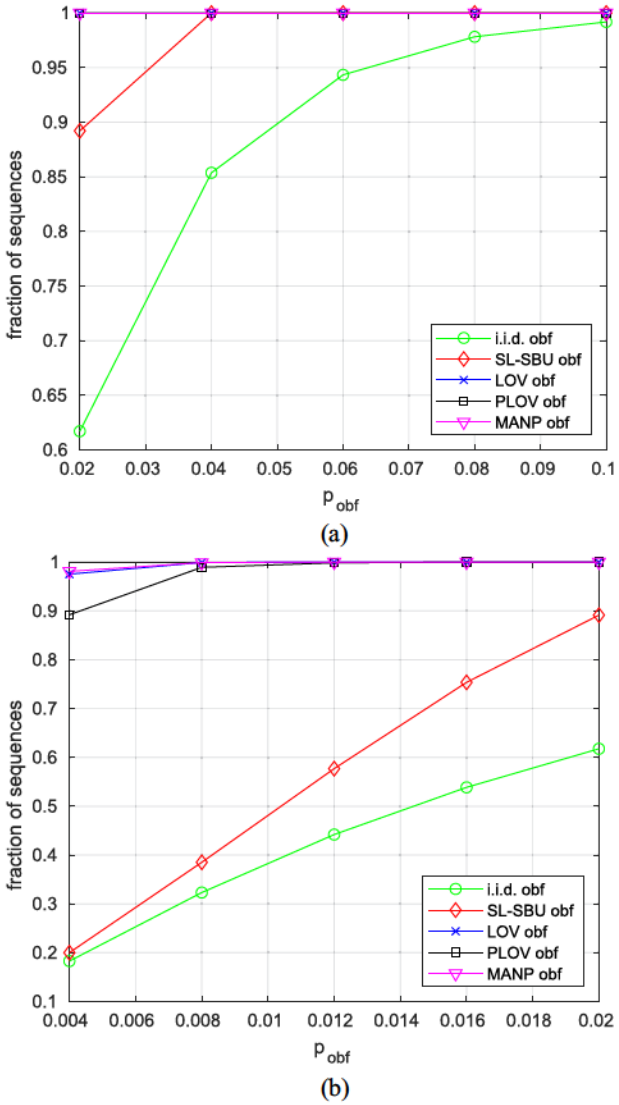

Fig. 7. Performance comparison of DDO methods (LOV, PLOV, MANP) and DIO methods (SL-SBU and i.i.d. obfuscation sequences) on i.i.d. sequences: the fraction of sequences which contain user 1's identifying pattern ($[r-l+1,\ldots,r-1,r]$). $r = 20+l$, $l = 1$, $m = 1000$, $h = 10$. (a) $p_{\text{obf}} = 0.02{:}0.02{:}0.1$. (b) $p_{\text{obf}} = 0.004{:}0.004{:}0.02$

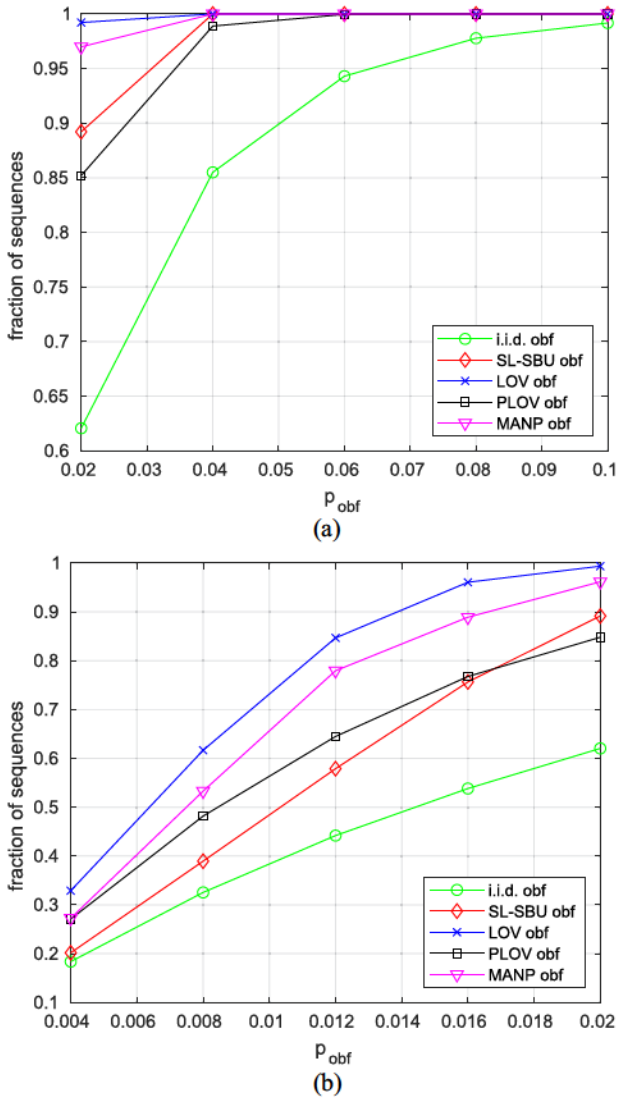

Fig. 8. Performance comparison of DDO methods (LOV, PLOV, MANP) and DIO methods (SL-SBU and i.i.d. obfuscation sequences) on the Reality Mining data set: the fraction of sequences which contain user 1's identifying pattern ($[r-l+1,\ldots,r-1,r]$). $r = 20+l$, $l = 1$, $m = 1000$, $h = 10$. Data points are sampled with interval at least 10 min. (a) $p_{\text{obf}} = 0.02{:}0.02{:}0.1$. (b) $p_{\text{obf}} = 0.004{:}0.004{:}0.02$.

the data does not provide enough variety for the DDO algorithms to create more unique patterns in comparison with the i.i.d. sequences.

We compare our proposed SL-SBU-based obfuscation method with the following existing privacy-protecting techniques: generalization, subsampling, and obfuscation with uniform noise, as follows.

A *generalization*-based privacy protecting method is formally employed on data sequences for preventing the private data (anonymized) being reidentified by linking QIDs with external information [50], [52], [53], [54]. Since the risk of users being identified by the QIDs, e.g., movement patterns of individuals or personal points of interest, will be reduced [85], generalization is a good benchmark defense method for pattern matching attacks. A generalization process is executed as: each data point $i \in \mathcal{R} = \{0, 1, \ldots, r-1\}$ is replace by $\lceil i/\text{group size} \rceil$, where the group size is the number of data points that each group contains (resolution). The total number of groups will be $\lceil r/\text{group size} \rceil$. Here, we assume all of the data points in the sequences will be generalized.

The data subsampling technique, which reduces sampling frequency, might potentially lower the reidentification risks by avoiding the presence of subsets of the data points [55], [56], [57], [58]. For obstructing pattern matching attacks, removing data points between the samplings might help create new patterns (activating some patterns by shortening the distance of two data points which are originally more widely spaced than h). The data point sequences are subsampled by the sampling period T_s, which means that the neighboring data points which appear in the sampled sequence are actually separated by at least time T_s.

We also consider obfuscation with uniform noise (employing i.i.d. obfuscation sequence), which has been simulated above.

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on December 15,2022 at 17:51:05 UTC from IEEE Xplore. Restrictions apply.

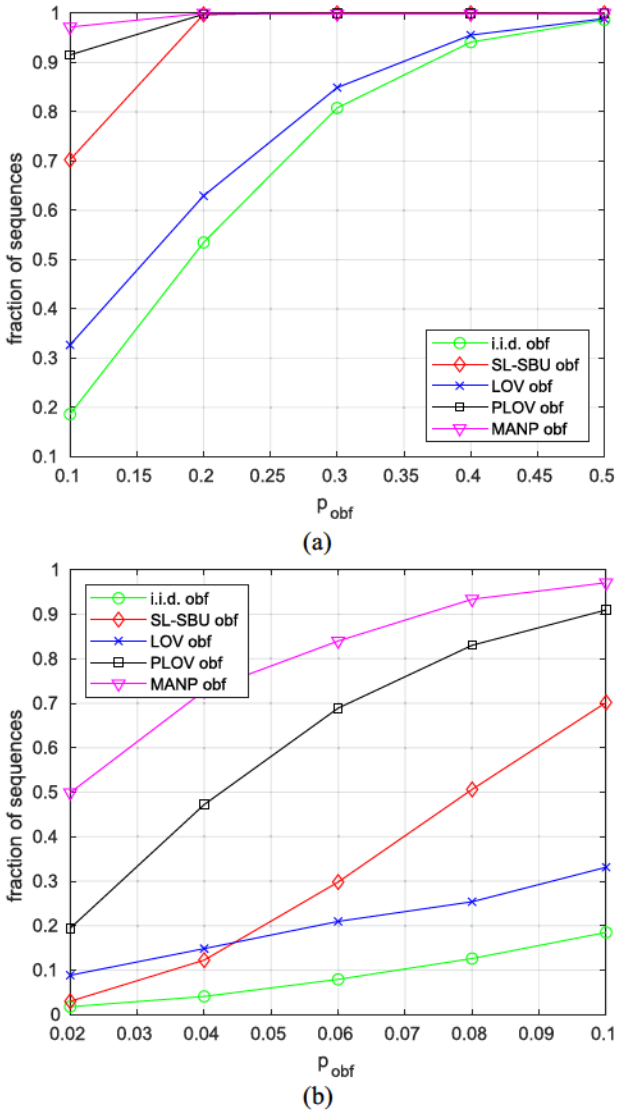

Fig. 9. Performance comparison of DDO methods (LOV, PLOV, MANP) and DIO methods (SL-SBU and i.i.d. obfuscation sequences) on i.i.d. sequences: the fraction of sequences which contain user 1's identifying pattern ($[r-l+1,\ldots,r-1,r]$). $r=20+l$, $l=2$, $m=1000$, $h=10$. (a) $p_{\text{obf}}=0.1{:}0.1{:}0.5$. (b) $p_{\text{obf}}=0.02{:}0.02{:}0.1$.

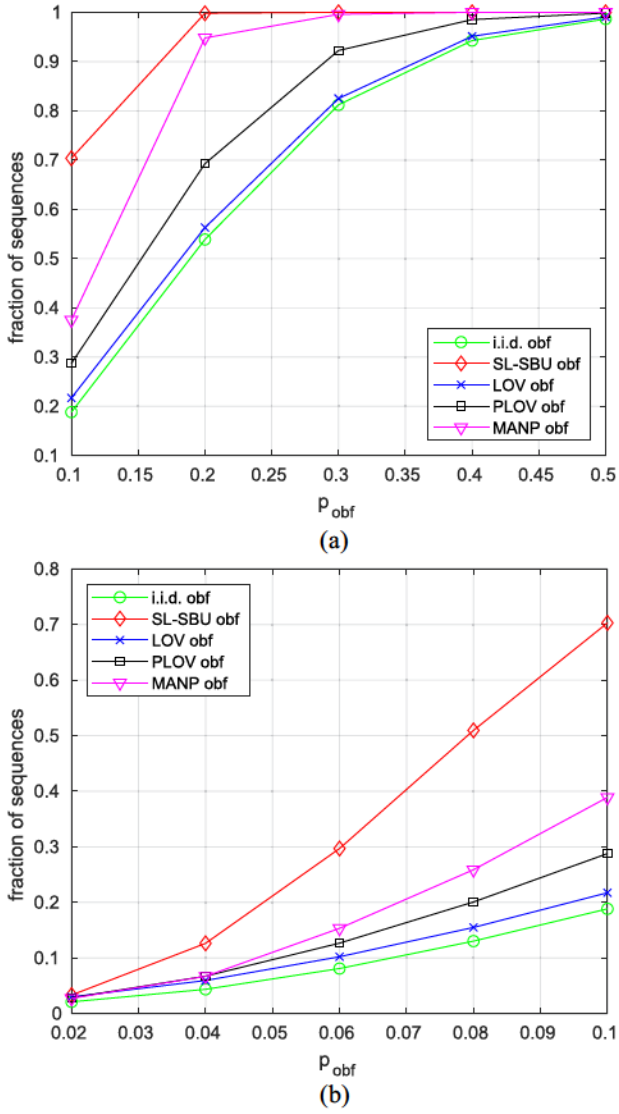

Fig. 10. Performance comparison of DDO methods (LOV, PLOV, MANP) and DIO methods (SL-SBU and i.i.d. obfuscation sequences) on the Reality Mining data set: the fraction of sequences which contain user 1's identifying pattern ($[r-l+1,\ldots,r-1,r]$). $r=20+l$, $l=2$, $m=1000$, $h=10$. Data points are sampled with interval at least 10 min. (a) $p_{\text{obf}}=0.1{:}0.1{:}0.5$. (b) $p_{\text{obf}}=0.02{:}0.02{:}0.1$.

Since the generalization and subsampling methods are unable to generate new letters outside the data point set $\mathcal{R}$, we validate their performances (fraction of users whose encoded sequence contains the identifying pattern) by averaging the results of generated random identifying patterns whose element letters are located inside $\mathcal{R}$. As shown by Tables VI and VII, for the generalization method, the privacy performance goes up with increasing group size, since increasing the group size boosts the chance of any sequence being replicated between two users due to the degradatation of the data resolution. For the subsampling method, increasing the sampling period T_s slightly helps improve the performance, though it might go down modestly at first due to the data points lost, and then go up afterwards with a continued reduction of the sampling rate. The performance of our proposed SL-SBU method achieves stable and considerable results in comparison with the existing methods. For instance, for pattern length $l=2$, the SL-SBU method can achieve around 50% of users with a pattern when applying the obfuscation with $p_{\text{obf}}=0.1$, while the generalization method achieves similar result by setting the group size $=3$ at the cost of degrading each sample in the data sequence. The subsampling method has the poorest performance, since it does not intend to create patterns as aggressively as the SL-SBU-based obfuscation method or the generalization method.

Utility: The relationship between utility and privacy (due to obfuscation) can vary in different application scenarios. Please note that our work is a generic study of IoT privacy through pattern matching, and we do not focus on any particular type of IoT system. However, as a rough analysis to demonstrate the PUT, one can assume a linear relationship between the utility cost and p_{obf}, since the ratio of obfuscated data points is a general measure of the loss of data usefulness [86], [87], [88], and especially it is obvious when the obfuscating data point is

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on December 15,2022 at 17:51:05 UTC from IEEE Xplore. Restrictions apply.

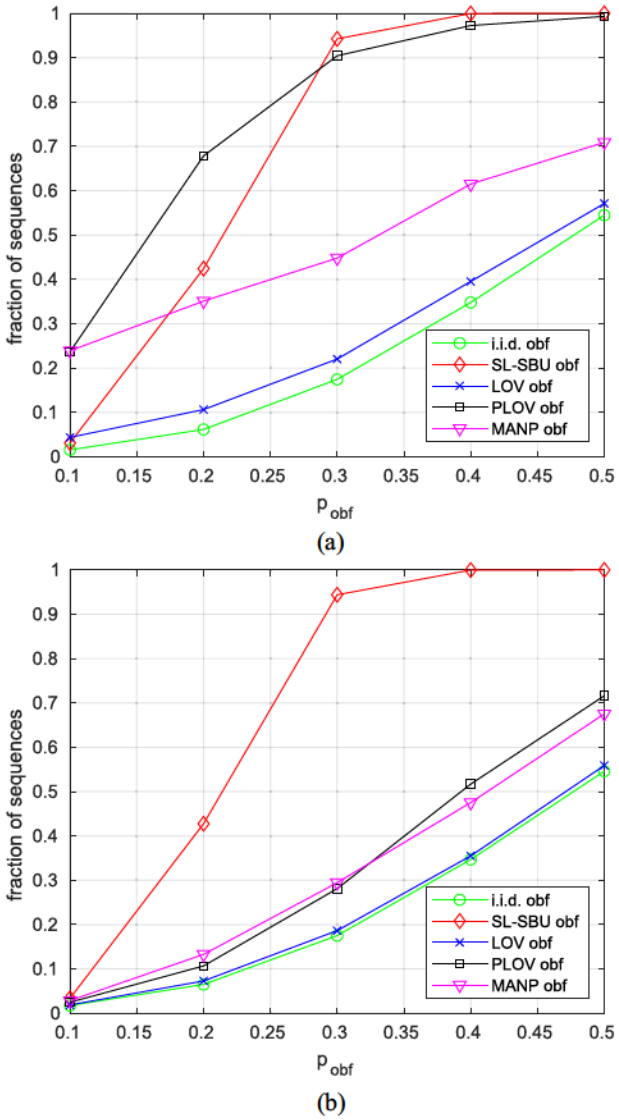

Fig. 11. Performance comparison of DDO methods (LOV, PLOV, MANP) and DIO methods (SL-SBU and i.i.d. obfuscation sequences) on i.i.d. sequences and Reality Mining sequences (data points are sampled with interval at least 10 min): the fraction of sequences which contain user 1's identifying pattern ($[r-l+1,\ldots,r-1,r]$). $r = 20+l$, $l = 3$, $m = 1000$, $h = 10$. (a) Performance on i.i.d. sequences. (b) Performance on Reality Mining sequences.

chosen uniformly randomly (even if not, we could still get the averaging utility cost by taking the expectation of the distortion in terms of each obfuscating data point and find it maps linearly to p_{obf}). However, in reality, the analysis of utility needs to be tailored to specific applications [89], and the relation becomes much more complicated due to many factors other than the data points, which is beyond the scope of this article.

IX. Conclusion

Various PPMs have been proposed to improve users' privacy in UDD services. To thwart pattern-matching attacks, we present data-independent and data-dependent PPMs that do not depend on a statistical model of users' data. In particular, a small noise is 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. We validate the proposed methods on both synthetic data and the Reality Mining data set to demonstrate their utility and compare their performance.

References

[1] B. Guan, N. Takbiri, D. L. Goeckel, A. Houmansadr, and H. Pishro-Nik, "Sequence obfuscation to thwart pattern matching attacks," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2020, pp. 884–889.

[2] F. Shirani, S. Garg, and E. Erkip, "Optimal active social network de-anonymization using information thresholds," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2018, pp. 1445–1449.

[3] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel, "A practical attack to de-anonymize social network users," in *Proc. IEEE Symp. Security Privacy*, 2010, pp. 223–238.

[4] G. Danezis and C. Troncoso, "Vida: How to use Bayesian inference to de-anonymize persistent communications," in *Proc. Int. Symp. Privacy Enhancing Technol. Symp.*, 2009, pp. 56–72.

[5] P.-M. Junges, J. François, and O. Festor, "Passive inference of user actions through IoT gateway encrypted traffic analysis," in *Proc. IFIP/IEEE Symp. Integr. Netw. Serv. Manag. (IM)*, 2019, pp. 7–12.

[6] N. Apthorpe, D. Reisman, S. Sundaresan, A. Narayanan, and N. Feamster, "Spying on the smart home: Privacy attacks and defenses on encrypted IoT traffic," 2017, *arXiv:1708.05044*.

[7] F. Staff, "Internet of Things: Privacy and security in a connected world," Federal Trade Commission, Washington, DC, USA, Rep., 2015. [Online]. Available: https://www.ftc.gov/news-events/events/2013/11/internet-things-privacy-security-connected-world

[8] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial Internet of Things," in *Proc. 52nd ACM/EDAC/IEEE Des. Autom. Conf. (DAC)*, 2015, pp. 1–6.

[9] H. Wang and F. P. Calmon, "An estimation-theoretic view of privacy," in *Proc. 55th Annu. Allerton Conf. Commun. Control Comput. (Allerton)*, 2017, pp. 886–893.

[10] A. Ukil, S. Bandyopadhyay, and A. Pal, "IoT-privacy: To be private or not to be private," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2014, pp. 123–124.

[11] M. Diaz, H. Wang, F. P. Calmon, and L. Sankar, "On the robustness of information-theoretic privacy measures and mechanisms," *IEEE Trans. Inf. Theory*, vol. 66, no. 4, pp. 1949–1978, Apr. 2020.

[12] B. Hoh and M. Gruteser, "Protecting location privacy through path confusion," in *Proc. 1st Int. Conf. Security Privacy Emerg. Areas Commun. Netw. (SecureComm)*, Athens, Greece, 2005, pp. 194–205.

[13] J. Unnikrishnan, "Asymptotically optimal matching of multiple sequences to source distributions and training sequences," *IEEE Trans. Inf. Theory*, vol. 61, no. 1, pp. 452–468, Jan. 2015.

[14] 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 Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1407–1418, Jun. 2019.

[15] F. M. Naini, J. Unnikrishnan, P. Thiran, and M. Vetterli, "Where you are is who you are: User identification by matching statistics," *IEEE Trans. Inf. Forensics Security*, vol. 11, pp. 358–372, 2016.

[16] Z. Montazeri, A. Houmansadr, and H. Pishro-Nik, "Achieving perfect location privacy in wireless devices using anonymization," *IEEE Trans. Inf. Forensics Security*, vol. 12, pp. 2683–2698, 2017.

[17] Y. Yoshida, M.-H. Yung, and M. Hayashi, "Optimal mechanism for randomized responses under universally composable security measure," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2019, pp. 547–551.

[18] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proc. 1st Int. Conf. Mobile Syst. Appl. Serv.*, San Francisco, CA, USA, 2003, pp. 31–42.

[19] C. A. Ardagna, M. Cremonini, E. Damiani, S. D. C. di Vimercati, and P. Samarati, "Location privacy protection through obfuscation-based techniques," in *Proc. DBSec*, 2007, pp. 47–60.

[20] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Proc. EUROCRYPT*, 2006, pp. 486–503.

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on December 15,2022 at 17:51:05 UTC from IEEE Xplore. Restrictions apply.

[21] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *J. Amer. Stat. Assoc.*, vol. 60, no. 309, pp. 63–69, 1965.

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

[23] N. Takbiri, A. Houmansadr, D. L. Goeckel, and H. Pishro-Nik, "Limits of location privacy under anonymization and obfuscation," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2017, pp. 764–768.

[24] S. Mangold and S. Kyriazakos, "Applying pattern recognition techniques based on hidden Markov models for vehicular position location in cellular networks," in *Proc. Gateway to 21st Century Commun. Village. (VTC-Fall) IEEE VTS 50th Veh. Technol. Conf.*, vol. 2, 1999, pp. 780–784.

[25] B.-H. Juang and L. R. Rabiner, "The segmental K-means algorithm for estimating parameters of hidden Markov models," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 9, pp. 1639–1641, Sep. 1990.

[26] F. Shirani, S. Gar, and E. Erkip, "A concentration of measure approach to database de-anonymization," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2019, pp. 2746–2752.

[27] D. Cullina, P. Mittal, and N. Kiyavash, "Fundamental limits of database alignment," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2018, pp. 651–655.

[28] O. E. Dai, D. Cullina, and N. Kiyavash, "Fundamental limits of database alignment," in *Proc. Mach. Learn. Res.*, 2019, pp. 651–655.

[29] N. Takbiri, R. Soltani, D. L. Goeckel, A. Houmansadr, and H. Pishro-Nik, "Asymptotic loss in privacy due to dependency in Gaussian traces," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Marrakech, Morocco, 2019, pp. 1–6.

[30] R. A. Becker *et al.*, "Route classification using cellular handoff patterns," in *Proc. 13th Int. Conf. Ubiquitous Comput.*, 2011, pp. 123–132.

[31] Q. Xu, A. Gerber, Z. M. Mao, and J. Pang, "AccuLoc: Practical localization of performance measurements in 3G networks," in *Proc. 9th Int. Conf. Mobile Syst. Appl. Serv.*, 2011, pp. 183–196.

[32] N. Eagle, J. A. Quinn, and A. Clauset, "Methodologies for continuous cellular tower data analysis," in *Proc. Int. Conf. Pervasive Comput.*, 2009, pp. 342–353.

[33] Y. Zhu, X. Fu, B. Graham, R. Bettati, and W. Zhao, "On flow correlation attacks and countermeasures in mix networks," in *Proc. Int. Workshop Privacy Enhancing Technol.*, 2004, pp. 207–225.

[34] M. Newey, "Notes on a problem involving permutations as subsequences," Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, Rep. CS-73-340, 1973.

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

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

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

[38] R. McPherson, R. Shokri, and V. Shmatikov, "Defeating image obfuscation with deep learning," 2016, *arXiv:1609.00408*.

[39] Q. Sun, L. Ma, S. J. Oh, L. Van Gool, B. Schiele, and M. Fritz, "Natural and effective obfuscation by head inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5050–5059.

[40] L. Fan, "Practical image obfuscation with provable privacy," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, 2019, pp. 784–789.

[41] X. Pan *et al.*, "FlowCog: Context-aware semantics extraction and analysis of information flow leaks in android apps," in *Proc. 27th USENIX Security Symp. (USENIX Security)*, 2018, pp. 1669–1685.

[42] X. Pan, Y. Cao, and Y. Chen, "I do not know what you visited last summer: Protecting users from third-party Web tracking with trackingfree browser," in *Proc. Annu. Netw. Distrib. Syst. Security Symp. (NDSS)*, San Diego, CA, USA, 2015, pp. 1–15.

[43] H. Lin and N. W. Bergmann, "IoT privacy and security challenges for smart home environments," *Information*, vol. 7, no. 3, p. 44, 2016.

[44] S. Zheng, N. Apthorpe, M. Chetty, and N. Feamster, "User perceptions of smart home IoT privacy," in *Proc. ACM Human-Comput. Interact.*, vol. 2, 2018, pp. 1–20.

[45] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: K-anonymity and its enforcement through generalization and suppression," Dept. Comput. Sci. Lab., SRI Int., Menlo Park, CA, USA, Rep. SRI-CSL-98-04, 1998.

[46] L. Sweeney, "*k*-anonymity: A model for protecting privacy," *Int. J. Uncertainty Fuzziness Knowl. Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002.

[47] Z. Tu, K. Zhao, F. Xu, Y. Li, L. Su, and D. Jin, "Protecting trajectory from semantic attack considering $k-$anonymity, $l-$diversity, and $t-$closeness," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 1, pp. 264–278, Mar. 2019.

[48] S. Zhang, X. Li, Z. Tan, T. Peng, and G. Wang, "A caching and spatial *K*-anonymity driven privacy enhancement scheme in continuous location-based services," *Future Gener. Comput. Syst.*, vol. 94, pp. 40–50, May 2019.

[49] J. Wang, Z. Cai, and J. Yu, "Achieving personalized *k*-anonymity-based content privacy for autonomous vehicles in CPS," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4242–4251, Jun. 2020.

[50] P. Samarati, "Protecting respondents identities in microdata release," *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 6, pp. 1010–1027, Nov./Dec. 2001.

[51] L. Sweeney, "Achieving *k*-anonymity privacy protection using generalization and suppression," *Int. J. Uncertainty Fuzziness Knowl. Based Syst.*, vol. 10, no. 5, pp. 571–588, 2002.

[52] S. Yaseen *et al.*, "Improved generalization for secure data publishing," *IEEE Access*, vol. 6, pp. 27156–27165, 2018.

[53] T. Kanwal *et al.*, "Privacy-preserving model and generalization correlation attacks for 1:M data with multiple sensitive attributes," *Inf. Sci.*, vol. 488, pp. 238–256, Jul. 2019.

[54] T. Kanwal, A. Anjum, and A. Khan, "Privacy preservation in e-health cloud: Taxonomy, privacy requirements, feasibility analysis, and opportunities," *Clust. Comput.*, vol. 24, no. 1, pp. 293–317, 2021.

[55] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Enhancing security and privacy in traffic-monitoring systems," *IEEE Pervasive Comput.*, vol. 5, no. 4, pp. 38–46, Oct.–Dec. 2006.

[56] Y. Xu, T. Ma, M. Tang, and W. Tian, "A survey of privacy preserving data publishing using generalization and suppression," *Appl. Math. Inf. Sci.*, vol. 8, no. 3, pp. 1103–1116, 2014.

[57] Y.-X. Wang, B. Balle, and S. P. Kasiviswanathan, "Subsampled Rényi differential privacy and analytical moments accountant," in *Proc. 22nd Int. Conf. Artif. Intell. Stat.*, 2019, pp. 1226–1235.

[58] B. Balle, G. Barthe, and M. Gaboardi, "Privacy profiles and amplification by subsampling," *J. Privacy Confidentiality*, vol. 10, no. 1, pp. 1–32, 2020.

[59] C. A. Ardagna, M. Cremonini, S. D. C. di Vimercati, and P. Samarati, "An obfuscation-based approach for protecting location privacy," *IEEE Trans. Dependable Secure Comput.*, vol. 8, no. 1, pp. 13–27, Jan./Feb. 2011.

[60] N. Takbiri, A. Houmansadr, D. L. Goeckel, and H. Pishro-Nik, "Privacy of dependent users against statistical matching," *IEEE Trans. Inf. Theory*, vol. 66, no. 9, pp. 5842–5865, Sep. 2020.

[61] H.-Y. Tran, J. Hu, and H. R. Pota, "Smart meter data obfuscation with a hybrid privacy-preserving data publishing scheme without a trusted third party," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 16080–16095, Sep. 2022.

[62] R. M. Karp, R. E. Miller, and A. L. Rosenberg, "Rapid identification of repeated patterns in strings, trees and arrays," in *Proc. 4th Annu. ACM Symp. Theory Comput.*, 1972, pp. 125–136.

[63] P. Weiner, "Linear pattern matching algorithms," in *Proc. 14th Annu. Symp. Switching Automata Theory (SWAT)*, 1973, pp. 1–11.

[64] B. Zhou, J. Pei, and W. Luk, "A brief survey on anonymization techniques for privacy preserving publishing of social network data," *ACM SIGKDD Explorations Newslett.*, vol. 10, no. 2, pp. 12–22, 2008.

[65] J. Unnikrishnan and F. M. Naini, "De-anonymizing private data by matching statistics," in *Proc. 51st Annu. Allerton Conf. Commun. Control Comput. (Allerton)*, 2013, pp. 1616–1623.

[66] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," *ACM SIGMOD Rec.*, vol. 23, no. 2, pp. 419–429, 1994.

[67] Q. Ma, B. Burns, K. Narayanaswamy, V. Rawat, and M. C. Shieh, "Network attack detection using partial deterministic finite automaton pattern matching," U.S. Patent 7 904 961, Mar. 8, 2011.

[68] S. Al-Khalifa, H. V. Jagadish, N. Koudas, J. M. Patel, D. Srivastava, and Y. Wu, "Structural joins: A primitive for efficient XML query pattern matching," in *Proc. 18th Int. Conf. Data Eng.*, 2002, pp. 141–152.

[69] M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama, and T. Koshiba, "Secure pattern matching using somewhat homomorphic encryption," in *Proc. ACM Workshop Cloud Comput. Security Workshop*, 2013, pp. 65–76.

[70] B. Wang, W. Song, W. Lou, and Y. T. Hou, "Privacy-preserving pattern matching over encrypted genetic data in cloud computing," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, 2017, pp. 1–9.

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on December 15,2022 at 17:51:05 UTC from IEEE Xplore. Restrictions apply.

[71] J. Baron, K. El Defrawy, K. Minkovich, R. Ostrovsky, and E. Tressler, "5pm: Secure pattern matching," in *Proc. Int. Conf. Security Cryptogr. Netw.*, 2012, pp. 222–240.
[72] R. Rahmat, F. Nicholas, S. Purnamawati, and O. S. Sitompul, "File type identification of file fragments using longest common subsequence (LCS)," *J. Phys. Conf. Ser.*, vol. 801, no. 1, p. 12054, 2017.
[73] R. J. Povinelli and X. Feng, "A new temporal pattern identification method for characterization and prediction of complex time series events," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 2, pp. 339–352, Mar./Apr. 2003.
[74] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.
[75] S. Isaacman, R. Becker, R. Cáceres, S. Kobourov, J. Rowland, and A. Varshavsky, "A tale of two cities," in *Proc. 11th Workshop Mobile Comput. Syst. Appl.*, 2010, pp. 19–24.
[76] R. Keralapura, A. Nucci, Z.-L. Zhang, and L. Gao, "Profiling users in a 3G network using hourglass co-clustering," in *Proc. 16th Annu. Int. Conf. Mobile Comput. Netw.*, 2010, pp. 341–352.
[77] M. Mano and Y. Ishikawa, "Anonymizing user location and profile information for privacy-aware mobile services," in *Proc. 2nd ACM SIGSPATIAL Int. Workshop Location Based Soc. Netw.*, 2010, pp. 68–75.
[78] N. Takbiri, A. Houmansadr, D. L. Goeckel, and H. Pishro-Nik, "Privacy against statistical matching: Inter-user correlation," in *Proc. Int. Symp. Inf. Theory (ISIT)*. Vail, CO, USA, 2018, pp. 1036–1040.
[79] N. de Bruijn, "A combinatorial problem," *Proc. Sect. Sci. Koninklijke Nederlandse Akademie van Wetenschappen te Amsterdam*, vol. 49, no. 7, pp. 758–764, 1946.
[80] F. S. Annexstein, "Generating de Bruijn sequences: An efficient implementation," *IEEE Trans. Comput.*, vol. 46, no. 2, pp. 198–200, Feb. 1997.
[81] S. M. Ross, *Stochastic Processes*. New York, NY, USA: Wiley, 1996.
[82] M. Christodoulakis, C. S. Iliopoulos, L. Mouchard, and K. Tsichlas, "Pattern matching on weighted sequences," in *Proc. Algorithms Comput. Methods Biochem. Evol. Netw. (CompBioNets)*, London, U.K., 2004, pp. 17–30.
[83] Y. Cheng, I. Izadi, and T. Chen, "Pattern matching of alarm flood sequences by a modified Smith–Waterman algorithm," *Chem. Eng. Res. Des.*, vol. 91, no. 6, pp. 1085–1094, 2013.
[84] N. Eagle, A. S. Pentland, and D. Lazer, "Inferring friendship network structure by using mobile phone data," *Proc. Nat. Acad. Sci.*, vol. 106, no. 36, pp. 15274–15278, 2009.
[85] C. Bettini, X. S. Wang, and S. Jajodia, "Protecting privacy against location-based personal identification," in *Proc. Workshop Secure Data Manag.*, 2005, pp. 185–199.
[86] M. A. Erdogdu and N. Fawaz, "Privacy-utility trade-off under continual observation," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2015, pp. 1801–1805.
[87] S. Haney, A. Machanavajjhala, J. M. Abowd, M. Graham, M. Kutzbach, and L. Vilhuber, "Utility cost of formal privacy for releasing national employer-employee statistics," in *Proc. ACM Int. Conf. Manag. Data*, 2017, pp. 1339–1354.
[88] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 968–979, May 2020.
[89] V. Shejwalkar, A. Houmansadr, H. Pishro-Nik, and D. Goeckel, "Revisiting utility metrics for location privacy-preserving mechanisms," in *Proc. 35th Annu. Comput. Security Appl. Conf.*, 2019, pp. 313–327.

Bo Guan (Student Member, IEEE) received the B.Eng. degree in electrical engineering from Northeastern University, Shenyang, China, in 2015, and the M.Sc. degree in electrical engineering from Northwestern University, Evanston, IL, USA, in 2017. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Massachusetts at Amherst, Amherst, MA, USA.

His current research studies focus on information privacy and security.

Nazanin Takbiri (Student Member, IEEE) received the B.S. degree from the University of Tehran, Tehran, Iran, in 2012, the M.S. degree from Boğaziçi University, Istanbul, Turkey, in 2016, and the Ph.D. degree from the University of Massachusetts at Amherst, Amherst, MA, USA, in 2019.

Afterward, she joined Qualcomm Inc., San Diego, CA, USA, where she was a Senior Security Researcher. She is currently a Senior Security Architect with Microsoft Corporation, Redmond, WA, USA. Her current research interests include privacy and security issues with a focus on the Internet of Things, post-quantum cryptography, and embedded systems security against passive and active physical attacks.

Dennis Goeckel (Fellow, IEEE) received the B.S. degree from Purdue University, West Lafayette, IN, USA, in 1992, and the M.S. and Ph.D. degrees from the University of Michigan, Ann Arbor, MI, USA, in 1993 and 1996, respectively.

Since 1996, he has been with the ECE Department, University of Massachusetts at Amherst, Amherst, MA, USA, where he is currently a Professor.

Prof. Goeckel received the University of Massachusetts Distinguished Teaching Award in 2007. He received the NSF CAREER Award in 1999 and is an IEEE Fellow for "contributions to wireless communication systems and networks." He was a Lilly Teaching Fellow from 2000 to 2001.

Amir Houmansadr (Member, IEEE) received the Ph.D. degree from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 2012.

He is currently an Associate Professor with the Manning College of Information and Computer Sciences, University of Massachusetts at Amherst, Amherst, MA, USA. He works on specific problems, such as privacy-enhancing technologies, adversarial machine learning, statistical traffic analysis, and covert communications. His broad area of research is network security and privacy.

Dr. Houmansadr has received several awards, including the Best Practical Paper Award at the IEEE Symposium on Security and Privacy in 2013, a Google Faculty Research Award in 2015, an NSF CAREER Award in 2016, and a DARPA Young Faculty Award in 2022.

Hossein Pishro-Nik (Member, IEEE) received the B.Sc. degree in electrical and computer engineering from the Sharif University of Technology, Tehran, Iran, in 2001, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from Georgia Institute of Technology, Atlanta, GA, USA, in 2003 and 2005, respectively.

He is currently a Professor with the Department of Electrical and Computer engineering, University of Massachusetts at Amherst, Amherst, MA, USA. His research interests include information theory, wireless networks, vehicular/UAV networks, privacy, statistical learning, and decision making.

Prof. Pishro-Nik's awards include an NSF Faculty Early Career Development (CAREER) Award, an Outstanding Junior Faculty Award from UMass, and an Outstanding Graduate Research Award from the Georgia Institute of Technology. He has served as an Associate Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON COMMUNICATIONS, and IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY.

Authorized licensed use limited to: University of Massachusetts Amherst. Downloaded on December 15,2022 at 17:51:05 UTC from IEEE Xplore. Restrictions apply.