Constrained Obfuscation to Thwart Pattern Matching Attacks

Saeede Enayati
Electrical and
Computer Engineering
UMass, Amherst
senayati@umass.edu

Dennis L. Goeckel
Electrical and
Computer Engineering
UMass, Amherst
dgoeckel@ecs.umass.edu

Amir Houmansadr
College of Information
and Computer Sciences
UMass, Amherst
amir@cs.umass.edu

Hossein Pishro-Nik
Electrical and
Computer Engineering
UMass, Amherst
pishro@ecs.umass.edu

Abstract—**Recently, we have proposed a model-free privacy-preserving mechanism (PPM) against attacks that compromise user privacy by matching patterns in data sequences to those that are unique to a given user [1]. Because the PPM is model-free, there are no requirements on the statistical model for the data, which is desirable when the model is not perfectly known. However, the proposed PPM did not enforce any constraints on the value to which a data point might be obfuscated, hence allowing an unlikely pattern that would make it easy for the adversary to detect which values have been obfuscated. In this paper, we consider a constrained PPM that enforces a continuity constraint so as to avoid abrupt jumps in the obfuscated data. To design such, we employ a graph-based analytical framework and the concept of consecutive patterns. At each point, the obfuscated data should be chosen strictly from that point's neighbors. Unfortunately, this might undesirably increase the noise level employed in data obfuscation and hence unacceptably reduce utility. We propose a new obfuscation algorithm, namely the obfuscation-return algorithm, and characterize its privacy guarantees under continuity and noise level constraints.**

Index Terms—**Obfuscation, privacy-preserving mechanism (PPM), pattern matching, consecutive pattern.**

I. Introduction

Data-driven services and applications improve the user experience by continuously collecting and analyzing users' personal data. The utility of the user data for the application relies on its accuracy and timeliness. However, as the accuracy and timeliness are improved, users' privacy becomes vulnerable, hence necessitating adopting effective privacy-preserving mechanisms (PPMs) against various threats and attacks [2]–[4].

An essential assumption underlying many privacy-preserving mechanisms is that the statistical model of the data is known to the privacy designers. In other words, the designers adopt a model-based approach where the data is assumed to follow a statistical model such as an independent and identically distributed (i.i.d.), Markov chain, or a given distribution function [5]–[9]. The model-based approach has a major limitation: it is limited to that specific model and, if the underlying model is different from the one considered, the privacy of users can be compromised [10].

A model-free approach where no assumptions have been made on the statistical model of the data was proposed in [1] and [11] to thwart pattern matching attacks, where the adversary identifies the user for a given data sequence by locating an identifying pattern that is known to him from a previous knowledge or observation [12]. Such pattern matching has been employed in scenarios such as fingerprinting webpages visited by users [13], [14], linking communicating parties on messaging applications [15], and inferring the activities of the users of IoT devices [16].

The idea in [1], [11] is to obfuscate the data sequences such that any identifying pattern appears in the data sequence for a large number of users. However, in these prior works, there is no constraint on the values produced by the obfuscation and hence the adversary may be able to identify values that have been obfuscated by noting anomalies in the data sequences, such as a large jump in location data from one time instance to the next. Consider the case where the data denoted by $X_u(k)$ is the location of user u at time k. Further, assume that the sampling rate is relatively high. In this case, $X_u(k)$ and $X_u(k+1)$ must show values that are geographically close to each other. Therefore, if we obfuscate $X_u(k+1)$ to a value that is far from $X_u(k)$, as is possible in the methods of [1], [11], the adversary can easily detect the presence of an obfuscation. To avoid this potential limitation, here we enforce consistency constraints on the obfuscated user data sequences, such as a continuity constraint requiring that adjacent sequence elements have similar values.

Therefore, in this paper, the goal is to extend our recent model-free approach to a constrained version. By using a graph-based framework and the concept of consecutive patterns, we apply a continuity constraint on the data obfuscation algorithm to make certain there are no abrupt jumps in the obfuscated data. In particular, at each point that we aim to obfuscate the pattern, we are required to choose the obfuscated points only from the neighbors of that point, and this constraint inevitably increases the noise level in the obfuscated data and hence potentially reduces user utility. To address this challenge, we introduce a new algorithm termed the "obfuscation-return" algorithm, and we demonstrate that it ensures users have common consecutive patterns with a probability greater than a certain threshold (privacy), while keeping the data

This work was supported by NSF under grants CNS-1932326 and CNS-1739462.

manipulation below a certain desired threshold (utility).

A. Related Work

To thwart privacy attacks, different kinds of privacy-preserving mechanisms that address the nature of the attack and data itself have been developed. The concept of $k-$anonymity was introduced in [17] and investigated further for location-based services, for example [18]–[21]. $k-$anonymity has been also investigated widely in social networks, which is not as straightforward as in rational data tables [22]–[27].

However, anonymization can be vulnerable to de-anonymization attacks that are based on previous knowledge of user's trajectory information and patterns [28], [29]. Hence, obfuscation methods where data perturbation techniques are employed are used to protect users' data [30]. In this regard, [5], [31] obtained the limits of location privacy for the i.i.d. and dependent users, respectively, whereas [32] proposed a location perturbation for adversaries with approximate background location knowledge. A comparison between encryption and obfuscation was investigated in [33] for location data in terms of latency and computational load. Anonymization and obfuscation techniques were investigated in [34] where user traces are i.i.d. Gaussian time series with a user-dependent mean.

For addressing pattern matching attacks, there are many studies in different scenarios for different types of data. For example, efficient privacy-preserving wildcard pattern matching for IoT data was investigated in [35]. Also, PPMs against matching attacks for ridesharing applications were developed in [36], and privacy-preserving string pattern matching in cloud databases was investigated in [37], [38].

B. Contributions and Organization

The contributions of this paper are as follows:

- We provide a constrained model-free privacy-preserving method in the presence of pattern matching attacks.
- Using tools from graph theory, we develop a new obfuscation algorithm in which the continuity of the obfuscated data is satisfied.
- We show that the proposed algorithm not only satisfies the continuity and noise level constraints, but also provides privacy guarantees.

This paper is organized as follows: In Section II, the system model is provided. Section III provides the proposed constrained obfuscation algorithm, and Section IV concludes the paper.

II. System Model

As noted in the Introduction, similar to [1] and [11], we do not assume a statistical model for the data. However, in contrast to [1] and [11], our problem is further complicated by assuming that the (obfuscated) user data sequences must satisfy some continuity constraints, as follows.

Data Model: We denote by $X_u(k)$ the data of user u at time k. We assume that $X_u(k)$ can take one of $r \geq 2$ possible values in $\mathcal{R} = \{0, 1, \dots, r-1\}$. Furthermore, the user data vector is denoted by $\mathbf{X}_u = [X_u(1), X_u(2), \dots, X_u(m)]^T$ which represents m data points for user u. To have a general setting, we model the constraint on the data using a (pseudo-)graph $G(V, E)$. Here $V = \mathcal{R} = \{0, 1, \dots, r-1\}$. The edges of the graph, $E \subset V \times V$, show the constraint on the data. Specifically, if $X_u(k) = i$, then $X_u(k+1)$ can take only values from the set $N(i) = \{j \in V : (i,j) \in E\}$ where $N(i)$ is the set of neighbors of Node i. For example, in the case of location data, $N(i)$ could show the locations that are geographically close to i. We use an undirected graph with the understanding that if $(i,j) \in E$, then $(j,i) \in E$. We also allow loops, i.e., $(i,i) \in E$. This could be interpreted as saying the user stays in location i. The degree of a vertex $v \in V$, shown as $deg(v)$, is the number of nodes that are connected to v, with the understanding that loops contribute only one unit to the degree of the associated nodes. We denote the maximum degree by d_{max}, so for any $v \in V$, we have $deg(v) \leq d_{max}$. The maximum degree is a fixed integer value which is a property of the graph G, so it is independent of the number of users or the number of adversary observations per user. The graph for user u, shown by $G_u(V_u, E_u)$ is a subgraph of G, i.e., $V_u \subset V$ and $E_u \subset E$.

We assume some regularity conditions, which can be viewed as thresholds for which a node or edge is included in the user graph, respectively:

1) If $i \in V_u$, we assume that the portion of times that $X_u(k) = i$ is bounded away from zero (otherwise, i would not have been included in V_u). Formally, there exists $\delta_1 > 0$ such that for all users u, we have

$$\text{For all } i \in V_u, \ \mathbb{P}\left(\liminf_{m \to \infty} \frac{|\{k : X_u(k) = i\}|}{m} \geq \delta_1\right) = 1.$$

2) Next, we assume that if $(i,j) \in E_u$, and $X_u(k) = i$, then the probability that $X_u(k+1) = j$ is bounded away from zero. Formally, there exists $\delta_2 > 0$ such that for all users u, we have

$$\begin{aligned}
&\text{If } (i,j) \in E_u, \text{then} \\
&\mathbb{P}\left(X_u(k+1) = j | X_u(k) = i, X_u(k-1) = i_{k-1},\right. \\
&\qquad\qquad \left. X_u(k-2) = i_{k-2}, \dots, X_u(1) = i_1\right) > \delta_2.
\end{aligned}$$

Adversary Model: We assume that for each user, the adversary has access to the m obfuscated data points. For pattern matching, we assume that the adversary aims at identifying a user's data sequence by matching a specific pattern for that user. Specifically, if $\mathbf{Q}_v = q_v^{(1)}, q_v^{(2)}, \dots, q_v^{(l)}$, where $l \geq 2$, is a pre-identified pattern of user v, the adversary looks for this pattern in the obfuscated sequences to identify user v. Finally, we assume that the adversary knows the obfuscation algorithm; however, he does not know the realization of the random elements of the obfuscation mechanism.

Constrained Obfuscation Mechanism: We now discuss requirements for the obfuscation. Let $\mathbf{Z}_u$ be the obfuscated sequence for user u. Remember our goal here is to obfuscate the data in a way that the obfuscated sequence has a large number of patterns. Having the context in mind (e.g., the data being locations visited consecutively), we define a consecutive pattern as follows

Definition 1. A *consecutive pattern* is a sequence $\mathbf{Q} = q^{(1)}q^{(2)}\cdots q^{(l)}$, where $(q^{(i)}, q^{(i+1)}) \in E$ for all $i \in \{1, 2, \cdots, l-1\}$. A user u is said to have the consecutive pattern $\mathbf{Q}$ if the sequence $\mathbf{Q}$ is a subsequence of user u's data sequence in a consecutive way.

However, here the problem is further complicated by the requirement that the sequence $Z_u(1), Z_u(2), \dots$ must be a valid sequence according to G. In general, we require that the obfuscation mechanism must satisfy the following properties:

1) The sequence $Z_u(1), Z_u(2), \dots$ must be a valid sequence according to G. That is, for any $k \in \{1, 2, \cdots\}$, we must have
$$(Z_u(k), Z_u(k+1)) \in E.$$
2) Data and utility constraints must be maintained below acceptable thresholds. Specifically, in the following, (a) means we cannot change too many values, and (b) means that we cannot change any value by too much.
   a) The noise level (which shows the portion of altered data points) must be below a set threshold. Formally, if we define
   $$A_m = \frac{|\{k \in \{1, 2, \cdots, m\} : Z_u(k) \neq X_u(k)\}|}{m},$$
   we would like
   $$\mathbb{P}(A_m > a(n)) \to 0, \text{ as } n \to \infty,$$
   for the set threshold $a(n)$ which is sufficiently small.
   b) Next, to ensure the utility of the data, we require that for all users u and all $k = 1, 2, 3, \cdots$, we must have
   $$d(X_u(k), Z_u(k)) \leq D,$$
   where $d(i, j)$ shows the distance of nodes i and j in G (the length of a shortest path from i to j), and D is the threshold set by the application.
3) Finally, we would like our obfuscation mechanism to create a large number of patterns. Specifically, let $\mathbf{Q} = q^{(1)}q^{(2)}\cdots q^{(l)}$ be an arbitrary consecutive pattern of length l on graph G. Let n be the number of users u for whom $q^{(1)} \in V_u$. We require that the probability that the obfuscated sequence of length m of user u has the consecutive pattern $\mathbf{Q}$, denoted by $\mathbb{P}(\mathcal{B}_u)$ to be larger than some $c(n, m)$. This property requires for m and n to be large enough, specifically, $m = \Omega(1)$ and $n = \Omega(1)$.

Definition 2. An obfuscation mechanism satisfying the above properties is said to be a $(G, a(n), D, c(n, m))-$obfuscation.

III. Constrained Obfuscation

We now provide a specific $(G, a(n), D, c(n, m))-$obfuscation algorithm, named **Obfuscate-Return (OR)** and prove it satisfies the required properties for a specific choice of $a(n), D$, and $c(n, m)$. For a finite set S, we define $rand(S)$ as a randomly chosen element from S, where all elements are equally likely, and the choice is independent of any other source of randomness in the problem. We also need to define an auxiliary random process, $W(k) \in \{0, 1, 2, \cdots, l-1\}$, where $W(1) = 0$, to follow the location of the obfuscation process. Before providing the OR algorithm, we introduce two operations that the algorithm uses repeatedly.

1) The first, is the obfuscation, shown as $Obf(k, p)$, in which p is a probability measure independent of any other random sources. In this operation:
   - With probability p:
   $$Z_u(k) = rand\left(N(Z_u(k-1))\right),$$
   $$W(k) = W(k-1) + 1.$$
   - With probability $1 - p$:
   $$Z_u(k) = X_u(k),$$
   $$W(k) = 0.$$

   In other words, at time k, $Obf(k, p)$ chooses an obfuscated data point from the set of neighbors of the point $Z_u(k-1)$ with probability p, or the original data point with probability $1 - p$. In the former we let $W(k) = W(k-1) + 1$, where in the latter $W(k) = 0$.
2) The second operation is called the return operation, shown as $Ret(k)$, which is defined as follows:
   $$Z_u(k) = rand(\underset{v \in N(Z_u(k-1))}{\arg\min} d(v, X_u(k))),$$
   $$W(k) = 0 \quad \text{if} \quad Z_u(k) = X_u(k).$$
   In this equation, the $\arg\min(.)$ determines the set of nodes v from $Z_u(k-1)$'s neighbors that have the minimum distance from $X_u(k)$, and $Ret(k)$ then chooses randomly one of these nodes in order to return to the original pattern. Finally, during the return process, when $Z_u(k) = X_u(k)$, we reset $W(k) = 0$.

We now provide the OR(l, p_{obf}) algorithm.

Obfuscate-Return (OR) Algorithm: Let $Z_u(1) = X_u(1)$. At time $k > 1$,

1) If $W(k-1) = 0$, then $Obf(k, p_{obf})$.
2) If $1 \leq W(k-1) < l-1$, then $Obf(k, 1)$.
3) If $W(k-1) = l-1$, then $Ret(k)$.

Intuitively, we start obfuscating at some point, say k, with probability p_{obf} if $W(k-1) = 0$ which indicates there was no obfuscation at $k-1$, and we continue to obfuscate for $l-1$ data points. Then the $Ret(k)$ is conducted to determine the last obfuscation node in order to return to the original path.

Now, before stating the main theorem, let's make a simple observation.

Lemma 1. For the OR algorithm, we have
$$d(X_u(k+b), Z_u(k+b)) \leq d(X_u(k), Z_u(k)) + 2b.$$
for any $k, b \in \{1, 2, 3, \cdots\}$.

Proof. At each stage of the algorithm, the distance can increase by at most 2, i.e,

$$d(X_u(k+1), Z_u(k+1)) \leq d(X_u(k), Z_u(k)) + 2.$$

The Lemma can then be proved by induction. □

For simplicity, let's define $D_k = d(X_u(k), Z_u(k))$, so we have $D_{k+b} \leq D_k + 2b$.

Theorem 1. Let $p_{obf} = \frac{1}{n^\theta}$, where $0 < \theta < 1$ is arbitrary. The OR(l, p_{obf}) algorithm defined above is a $(G, a(n), D, c(n, m))-$obfuscation method, where

1) $D = 2(l-1)$;
2) $a(n) = \frac{c_1}{n^{\theta-\gamma}}$, where $c_1 > 0$ is a constant independent of n, and γ is an arbitrarily small constant independent of n.
3) $\mathbb{P}(\mathcal{B}_u) \geq c(n, m) = \frac{c_2}{n^\theta}$, for some constant c_2 independent of $n = \Omega(1)$ and $m = \Omega(1)$.

Proof. 1) First, we prove that the sequence $\{Z_u(k)\}_{k=1}^{\infty}$ is a valid sequence according to G, i.e., for any $k > 1$, we have $(Z_u(k-1), Z_u(k)) \in E$. We consider three cases:

a) If $W(k-1) = 0$, then $Z_u(k-1) = X_u(k-1)$. In this case, the algorithm runs the $Obf(k, p_{obf})$ operation. Which means either $Z_u(k) = X_u(k)$ so $(Z_u(k-1), Z_u(k)) = (X_u(k-1), X_u(k))$, or $Z_u(k) \in N(Z_u(k-1))$.

b) If $1 \leq W(k-1) < l-1$, then $Obf(k, 1)$ is run, which means $Z_u(k) \in N(Z_u(k-1))$.

c) If $W(k-1) = l-1$, then $Ret(k)$ is executed, which satisfies $Z_u(k) \in N(Z_u(k-1))$.

2) Next, we investigate the two utility constraints:

a) Here, we show that the noise level is below $\frac{c_1}{n^\theta}$. Define

$$T_m = |\{k \in \{1, 2, \cdots, m\} : Z_u(k) \neq X_u(k)\}|,$$

so $A_m = \frac{T_m}{m}$. Our goal is to show that T_m cannot be very large. Remember, $Z_u(1) = X_u(1)$. Let

$$K_1 = \min\{k > 1 : W(k) = 1\},$$

$$K_1' = \min\{k > K_1 : W(k) = 0\}.$$

For $j > 1$, define

$$K_j = \min\{k > K_{j-1}' : W(k) = 1\},$$

$$K_j' = \min\{k > K_j : W(k) = 0\}.$$

Also, let $N(m)$ be the number of K_js, i.e.,

$$N(m) = \max\{j : K_j < m\}.$$

Then, we can write

$$T_m = \sum_{j=1}^{N(m)} (K_j' - K_j) = \sum_{j=1}^{N(m)} T_{m,j},$$

where $T_{m,j} = K_j' - K_j$. Note that $T_{m,1}, T_{m,2}, T_{m,3}, \cdots$ are i.i.d, and also $N(m)$ is a stopping time for the random process $T_{m,1}, T_{m,2}, T_{m,3}, \cdots$, which means that we can use Wald's Equation [39] to conclude

$$E[T_m] = E\left[\sum_{j=1}^{N(m)} T_{m,j}\right] = E[N(m)]E[T_{m,j}]. \qquad (1)$$

Lemma 2. For all $j = 1, 2, \cdots, N(m)$, we have

$$E[T_{m,j}] \leq (l-1)\left(1 + \frac{1}{\delta_2}\right).$$

Proof. Investigating the OR algorithm, we observe that at any K_j, the $Obf(K_j, p_{obf})$ has been executed with the result $Z_u(k)$ being chosen randomly from the set $N(Z_u(k-1))$. We note that $K_j' \geq K_j + l - 1$ and

$$D_{K_j+l-2} = d(X_u(K_j+l-2), Z_u(K_j+l-2)) \leq 2(l-1).$$

This can be concluded from Lemma 1 and the fact that $Z_u(K_j-1) = X_u(K_j-1)$. Also note that for all $k \in \{K_j+l-1, K_j+l, \cdots, K_j'\}$, the $Ret(k)$ operation is executed. We now claim that for all $k \in \{K_j+l-1, K_j+l, \cdots, K_j'\}$, we have $\mathbb{P}(D_{k+1} \leq D_k - 2) > \delta_2$. To see this, consider the shortest path between $Z_u(k)$ and $X_u(k) = i$, say $Z_u(k), v_1, v_2, \cdots, v_{D_k-1}, X_u(k)$. Then, we know that

$$\mathbb{P}\left(X_u(k+1) = v_{D_k-1} | X_u(k) = i, X_u(k-1) = i_{k-1},\right.$$
$$\left. X_u(k-2) = i_{k-2}, X_u(1) = i_1\right) > \delta_2.$$

This means that if the $Ret(k)$ operation chooses $Z_u(k+1) = v_1$, then $D_{k+1} = D_k - 2$. Since the $Ret(k)$ operation chooses $Z_u(k+1)$ in a way to minimize D_{k+1}, we conclude that

$$\mathbb{P}(D_{k+1} \leq D_k - 2) > \delta_2.$$

Therefore, we can write

$$T_{m,j} \leq l - 1 + F_j,$$

where F_j is a Pascal random variable with parameters $l-1$ and δ_2 (we adopt the definition where F_j is the total number of successes and failures, so $F_j > 0$). Therefore,

$$\begin{aligned} E[T_{m,j}] &\leq l - 1 + E[F_j] \\ &= l - 1 + \frac{l-1}{\delta_2} \\ &= (l-1)\left(1 + \frac{1}{\delta_2}\right), \end{aligned}$$

which completes the proof. □

Lemma 3. We have

$$E[N(m)] \leq \frac{m}{n^\theta}.$$

Proof. This follows immediately from the algorithm definition. We can think of $N(m)$ as a counting process. Each new K_j (arrival) happens when an obfuscation occurs with probability $p_{obf} = \frac{1}{n^\theta}$. However we cannot have any arrivals between K_j and K_j'. Therefore, the number of arrivals here is less than or equal to the number of arrivals for a corresponding $Bernoulli(p_{obf})$ process (where arrivals could also happen between K_j and K'_j). We conclude that

$$E[N(m)] \leq mp_{obf} = \frac{m}{n^{\theta}}.$$

□

Using (1), Lemma 2, and Lemma 3, we conclude that

$$E[T_m] \leq (l-1)\left(1+\frac{1}{\delta_2}\right)\frac{m}{n^{\theta}}. \qquad (2)$$

Since $A_m = \frac{T_m}{m}$, by letting $a(n) = \frac{c_1}{n^{\theta-\gamma}}$, we conclude

$$\begin{aligned}
\mathbb{P}\left(A_m > a(n)\right) &= \mathbb{P}(T_m > ma(n)) \\
&\leq \frac{E[T_m]}{ma(n)} \quad \text{(Markov's Inequality)} \\
&\leq (l-1)\left(1+\frac{1}{\delta_2}\right)\frac{m}{n^{\theta}ma(n)} \quad \text{(Eq. 2)} \\
&= c_1\frac{1}{n^{\gamma}} \to 0, \quad \text{as } n \to \infty,
\end{aligned}$$

where c_1 is a constant.

b) Next, we show that for all users u and all $k = 1, 2, 3, \cdots$, we have

$$D_k = d(X_u(k), Z_u(k)) \leq 2(l-1).$$

Examining the OR(l, p_{obf}) algorithm, we have $D_k = d(X_u(k), Z_u(k)) = 0$ for values of k outside of the $[K_j, K'_j - 1]$. Within each $[K_j, K'_j - 1]$, we apply Lemma 1 to obtain $D_k \leq 2(l-1)$, for all $k \leq K_j + l - 2$. For values of k, $K_j + l - 1 \leq k \leq K'_j - 1$ the $Ret(k)$ operation is performed and as we saw above, D_k will be decreasing. This proves that $D_k \leq 2(l-1)$ for all $k = 1, 2, 3, \cdots$.

3) Finally, we can compute $\mathbb{P}(\mathcal{B}_u)$ as follows. Let $\mathbf{Q} = q^{(1)}q^{(2)}\cdots q^{(l)}$ be an arbitrary consecutive pattern of length l on graph G. Consider a user u for whom $q^{(1)} \in V_u$. By our regularity assumption, we have

$$\mathbb{P}\left(\liminf_{m\to\infty}\frac{|\{k : X_u(k) = q^{(1)}\}|}{m} \geq \delta_1\right) = 1.$$

Since $m = \Omega(1)$, we conclude that with a probability converging to 1 (as $m \to \infty$), we have

$$|\{k : X_u(k) = q^{(1)}\}| = |I_u| \geq m\delta_1.$$

Now, define J_u as the set of integers $k \in I_u$, at which $Obf(k, p_{obf})$ operation is executed in the OR(l, p_{obf}) algorithm. We have

$$|J_u| \geq |I_u| - T_m.$$

Note that with probability converging to 1, we have $T_m \leq ma(n)$. We conclude that we have

$$|J_u| \geq \delta_1 m - ma(n).$$

Noting that $a(n) \to 0$ as $m \to \infty$, we conclude that for a constant $\delta'_1 > 0$, we have $|J_u| \geq \delta'_1 m$. In other words, for at least $\delta'_1 m$ values of k, we have $X_u(k) = q^{(1)}$, and the $Obf(k, p_{obf})$ operation is executed. Now for such values of k, we have

$$\begin{aligned}
&\mathbb{P}\left(Z_u(k+1) = q^{(2)}, Z_u(k+2) = q^{(3)}, \cdots, Z_{k+l-1} = q^{(l)}\right) \\
&= \mathbb{P}\left(Z_u(k+1) = q^{(2)}\right) \\
&\times \mathbb{P}\left(Z_u(k+2) = q^{(3)} | Z_u(k+1) = q^{(2)}\right) \\
&\cdots \times \mathbb{P}\left(Z_u(k+l-1) = q^{(l)} | Z_u(k+l-2) = q^{(l-1)}, \cdots,\right. \\
&\hspace{14em}\left. Z_u(k+1) = q^{(2)}\right) \\
&\hspace{8em}\overset{(a)}{\geq} \frac{p_{obf}}{d_{max}^{l-1}} = c_2\frac{1}{n^{\theta}},
\end{aligned}$$

where (a) comes from the fact that the probability of choosing a consecutive pattern's point from the neighbors of a current point is always greater than $\frac{1}{d_{max}}$ and c_2 is a constant. Thus, we can confirm $\mathbb{P}(\mathcal{B}_u) \geq \frac{c_2}{n^{\theta}}$, for some constant c_2 independent of n.

□

Although the results of this paper are obtained asymptotically where $m, n \to \infty$, they can be extended for finite m which is of our future interest.

IV. Conclusion

In our previous work [1], we proposed a model-free privacy-preserving method and showed we could use superstrings to preserve privacy against pattern matching attacks. Here, we perform the challenging extension of [1] to develop a constrained model-free obfuscation, which requires a significantly different approach. To this end, we enforced a continuity constraint on the obfuscated data so that the obfuscated data does not have any abrupt jumps or discontinuities that would indicate to the adversary that obfuscation had occurred. However, the continuity constraint makes the problem quite complicated. We employed concepts from graph theory and defined consecutive patterns to design a PPM for the constrained obfuscation problem. Analytical results show that the proposed algorithms satisfies privacy guarantees as well as the continuity and noise level constraints.

There are many avenues for future research. First, there could be "second order" characteristics of sequences that could result in leakage. For example, there might be information in the number of times that a specific pattern is repeated. In this case, not only do we need to ensure the occurrence of patterns in many sequences but also we need to ensure that they are repeated enough times in a fraction of the users. Second, one can assume a stronger adversary that has knowledge of a large number of different patterns for each user. In this case, it is essentially required to guarantee a large number of matchings in the graph. Finally, our obfuscation was independent of users' data. One can consider a data-dependent obfuscation where the key idea will be to choose obfuscated values in a way that at each point, the goal is to maximize the number of distinct patterns in the data sequence of user u based on the obfuscated sequence of the user so far.

References

[1] B. Guan, N. Takbiri, D. L. Goeckel, A. Houmansadr, and H. Pishro-Nik, "Sequence obfuscation to thwart pattern matching attacks," in *2020 IEEE International Symposium on Information Theory (ISIT)*, Los Angles, CA, USA, June 2020, pp. 884–889.

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

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

[4] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A survey on the internet of things (IoT) forensics: Challenges, approaches, and open issues," *IEEE Communications Surveys Tutorials*, vol. 22, no. 2, pp. 1191–1221, Secondquarter 2020.

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

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

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

[8] W. Wang, L. Ying, and J. Zhang, "On the relation between identifiability, differential privacy, and mutual-information privacy," *IEEE Transactions on Information Theory*, vol. 62, no. 9, pp. 5018–5029, Sept. 2016.

[9] B. Rassouli and D. Gündüz, "Optimal utility-privacy trade-off with total variation distance as a privacy measure," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 594–603, Mar. 2019.

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

[11] B. Guan, N. Takbiri, D. Goeckel, A. Houmansadr, and H. Pishro-Nik, "Superstring-based sequence obfuscation to thwart pattern matching attacks," *arXiv preprint arXiv:2108.12336, Under revision in IEEE Internet of Things Journal*, Aug. 2021.

[12] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore art thou R3579X? anonymized social networks, hidden patterns, and structural steganography," in *Proceedings of the 16th international conference on World Wide Web*, May 2007, pp. 181–190.

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

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

[15] G. Danezis and C. Troncoso, "Vida: How to use bayesian inference to de-anonymize persistent communications," in *International Symposium on Privacy Enhancing Technologies Symposium*. Springer, 2009, pp. 56–72.

[16] P.-M. Junges, J. François, and O. Festor, "Passive inference of user actions through IoT gateway encrypted traffic analysis," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2019, pp. 7–12.

[17] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[18] V. Sharma and C.-C. Shen, "Evaluation of an entropy-based k-anonymity model for location based services," in *International Conference on Computing, Networking and Communications (ICNC)*, Garden Grove, CA, USA, 2015, pp. 374–378.

[19] J. Wang, Y. Li, D. Yang, H. Gao, G. Luo, and J. Li, "Achieving effective k-anonymity for query privacy in location-based services," *IEEE Access*, vol. 5, pp. 24 580–24 592, Oct. 2017.

[20] N. Takbiri, A. Houmansadr, D. Goeckel, and H. Pishro-Nik, "Fundamental limits of location privacy using anonymization," in *51st Annual Conference on Information Science and Systems (CISS)*, Baltimore, MD, USA, 2017.

[21] F. Li, Y. Chen, B. Niu, Y. He, K. Geng, and J. Cao, "Achieving personalized k-anonymity against long-term observation in location-based services," in *IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6.

[22] B. Zhou and J. Pei, "Preserving privacy in social networks against neighborhood attacks," in *IEEE 24th International Conference on Data Engineering*, 2008, pp. 506–515.

[23] N. Li, N. Zhang, and S. K. Das, "Relationship privacy preservation in publishing online social networks," in *IEEE Third International Conference on Privacy, Security, Risk and Trust and IEEE Third International Conference on Social Computing*, Boston, MA, USA, Oct. 2011, pp. 443–450.

[24] M. Yuan, L. Chen, P. S. Yu, and T. Yu, "Protecting sensitive labels in social network data anonymization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 3, pp. 633–647, March 2013.

[25] P. Zhao, H. Jiang, C. Wang, H. Huang, G. Liu, and Y. Yang, "On the performance of k -anonymity against inference attacks with background information," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 808–819, Feb. 2019.

[26] H. Li, L. Gong, B. Wang, F. Guo, J. Wang, and T. Zhang, "k-anonymity based location data query privacy protection method in mobile social networks," in *International Conference on Networking and Network Applications (NaNA)*, Haikou City, China, Dec. 2020, pp. 326–334.

[27] L. Xing, X. Jia, J. Gao, and H. Wu, "A location privacy protection algorithm based on double k-anonymity in the social internet of vehicles," *IEEE Communications Letters*, vol. 25, no. 10, pp. 3199–3203, Oct. 2021.

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

[29] Y. Jian, S. Chen, Z. Zhang, and L. Zhang, "A novel scheme for protecting receiver's location privacy in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 10, pp. 3769–3779, Oct. 2008.

[30] C. A. Ardagna, M. Cremonini, S. De Capitani di Vimercati, and P. Samarati, "An obfuscation-based approach for protecting location privacy," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 1, pp. 13–27, Jan. 2011.

[31] N. Takbiri, A. Houmansadr, D. Goeckel, and H. Pishro-Nik, "Privacy of dependent users against statistical matching," *IEEE Transactions on Information Theory*, vol. 66, no. 9, pp. 5842–5865, Sept. 2020.

[32] R. Dewri, "Local differential perturbations: Location privacy under approximate knowledge attackers," *IEEE Transactions on Mobile Computing*, vol. 12, no. 12, pp. 2360–2372, 2013.

[33] V. A. Kachore, J. Lakshmi, and S. Nandy, "Location obfuscation for location data privacy," in *2015 IEEE World Congress on Services*, New York, NY, USA, 2015, pp. 213–220.

[34] K. Li, H. Pishro-Nik, and D. L. Goeckel, "Privacy against matching under anonymization and obfuscation in the gaussian case," in *52nd Annual Conference on Information Sciences and Systems (CISS)*, Princeton, NJ, USA, 2018, pp. 1–6.

[35] H. Qin, H. Wang, X. Wei, L. Xue, and L. Wu, "Privacy-preserving wildcards pattern matching protocol for IoT applications," *IEEE Access*, vol. 7, pp. 36 094–36 102, Feb. 2019.

[36] P. Hallgren, C. Orlandi, and A. Sabelfeld, "Privatepool: Privacy-preserving ridesharing," in *IEEE 30th Computer Security Foundations Symposium (CSF)*, Santa Barbara, CA, USA, Aug. 2017, pp. 276–291.

[37] B. Bezawada, A. X. Liu, B. Jayaraman, A. L. Wang, and R. Li, "Privacy preserving string matching for cloud computing," in *IEEE 35th International Conference on Distributed Computing Systems*, Columbus, OH, USA, July 2015, pp. 609–618.

[38] M. He, J. Zhang, G. Zeng, and S. M. Yiu, "A privacy-preserving multi-pattern matching scheme for searching strings in cloud database," in *15th Annual Conference on Privacy, Security and Trust (PST)*, Calgary, AB, Canada, Aug 2017.

[39] M. Roters, "On the validity of wald's equation," *Journal of applied probability*, vol. 31, no. 4, pp. 949–957, 1994.