CASTLEGUARD: Anonymised Data Streams with Guaranteed Differential Privacy

Alistair Robinson, Frederick Brown, Nathan Hall,
Alex Jackson, Graham Kemp, and Matthew Leeke
Department of Computer Science,
University of Warwick, Coventry, UK
E-mail: matthew.leeke@warwick.ac.uk

***Abstract*—Data streams are commonly used by data controllers to outsource the processing of real-time data to third-party data processors. Data protection legislation and best practice in data management support the view that data controllers are responsible for providing a guarantee of privacy for user data contained within published data streams. *Continuously Anonymising STreaming data via adaptive cLustEring* (CASTLE) is an established method for anonymising data streams with a guarantee of k-anonymity. However, k-anonymity has been shown to be a weak privacy guarantee that has vulnerabilities in practical applications. In this paper we propose *Continuously Anonymising STreaming data via adaptive cLustEring with GUARanteed Differential privacy* (CASTLEGUARD), a data stream anonymisation algorithm that provides a reliable guarantee of k-anonymity, l-diversity and differential privacy to data subjects. We analyse CASTLEGUARD to show that, through safe k-anonymisation and β-sampling, the proposed approach satisfies differentially private k-anonymity. Further, we demonstrate the efficacy of the approach in the context of machine learning, presenting experimental analysis to demonstrate that it can be used to protect the individual privacy of users whilst maintaining the utility of a data stream.**

***Keywords*-Privacy; Data Streams; Differential Privacy;**

I. Introduction

In an increasingly data-driven world, data controllers and processors must be able to provide a guarantee of privacy to comply with data protection legislation and meet the expectations of data subjects [1]. Data controllers, from online marketplace providers to government bodies, may wish to outsource the processing of data collected from the provision of their service to third-party data processors by publishing a data stream [2]. However, even in cases where uniquely identifying attributes are hidden, it has been shown that the privacy of a data subject can be compromised through a linkage attack by corroborating quasi-identifying (QI) attributes with publicly available data sets [3], [4], [5]. The exploitation of such vulnerabilities would violate the desire for privacy on the part of data subjects, who are increasingly aware of the risks of data collection and the importance of privacy when providing their data [6].

Guaranteeing k-anonymity [7] is a commonly used method of protecting against linkage and other privacy-focused attacks on a data set. In this context k-anonymity is defined as:

Definition (k**-anonymity).** *A data set S satisfies k-anonymity with respect to a set of quasi-identifiers QI if every generalisation of QI appears at least k times.*

Continuously Anonymising STreaming data via adaptive cLustEring (CASTLE) was proposed as a method of dynamically enforcing k-anonymity constraints on data streams [8]. Despite CASTLE achieving its stated aim of dynamically enforcing k-anonymity, it provides no privacy guarantees beyond k-anonymity. It has been shown that k-anonymity is not a strong privacy guarantee in many circumstances, leading to the exploration of stronger privacy guarantees, such as l-diversity and differential privacy [9], [10], [11]. As such, k-anonymity alone may be considered an insufficient privacy guarantee for data controllers and privacy-conscious data subjects.

In this paper we consider the scenario where a data controller wishes to publish a data stream to third-party data processors but requires a provable guarantee of privacy that is stronger than k-anonymity. Specifically, we propose *Continuously Anonymising STreaming data via adaptive cLustEring with GUARanteed Differential privacy* (CASTLEGUARD), a data stream anonymisation approach that provides a reliable guarantee of k-anonymity, l-diversity and differential privacy based on parameters l, β and ϕ. CASTLEGUARD achieves differential privacy for data streams by sampling entries from an input data stream S with probability β and using additive noise taken from a Laplace distribution with $\mu = 0$, $b = \frac{R}{\phi}$ where R is the range of an attribute.

A. Contributions

In this paper, we make the following specific contributions:

- We propose CASTLEGUARD, a data stream anonymisation approach that revises CASTLE to provide stronger privacy guarantees for data processors and subjects.
- We provide analysis to prove that CASTLEGUARD satisfies k-anonymity, l-diversity and differential privacy in a non-interactive model [12] and protects user privacy under a knowledgeable adversary.
- We evaluate the performance of CASTLEGUARD in the context of machine learning, demonstrating the efficacy of the approach by showing that information and data quality loss arising from anonymisation is sufficiently low to provide utility to data processors.

The overarching contribution of this paper is to demonstrate the applicability of non-interactive differential privacy to k-anonymity and l-diversity in the context of data streams.

B. Paper Structure

The remainder of this paper is structured as follows. In Section II we provide a brief survey of related work. In Section III we outline the assumed models. In Section IV we propose CASTLEGUARD, presenting the substantial outcomes of this paper. In Section VI we present the results of our analysis of CASTLEGUARD before Section VII concludes.

II. Related Work

In this section we provide a brief survey of research relating to privacy in data streams. This coverage focuses on the privacy issues addressed by CASTLEGUARD and the definitions of the properties it guarantees.

A. Data Streams

Data streaming is commonly employed by data controllers as a means to outsource data analysis to data processors [2], [13]. A data stream is modelled as an append-only sequence of tuples with an incremental ordering. It is more challenging for data controllers to provide privacy guarantees in the context of data streams than static data sets, not least because data streams have a temporal dimension and unknown size. Furthermore, the distribution of streamed data is likely, if not guaranteed, to change over time in most applications. Given these challenges, an adaptive solution is necessary to provide user privacy in data streams. For consistency of extension and comparison between CASTLE and CASTLEGUARD, we adopt our model of a data stream from [8]:

Definition (**Data Streams**). *A data stream S has schema $S(pid,\ QI,\ a_s)$ where pid is the unique identifier of the user primarily associated with the tuple, QI represents a tuple's set of quasi-identifiable attributes (Section I) and a_s represents a tuple's sensitive attribute (Section II-C). We consider S' as the anonymised form of a data stream S.*

B. k-anonymity in Data Streams

It can be difficult to use k-anonymity [7] in data stream anonymisation because of the challenges identified above. However, k-anonymity has been adapted for data streams in the development of CASTLE [8] and CASTLEGUARD:

Definition (k**-anonymised Data Streams**). *Consider a data stream $S(pid,\ QI,\ a_s)$. An anonymised data stream S' satisfies k-anonymity over QI if and only if every generalisation of QI appears at least k times in S'.*

CASTLE is an established anonymisation approach that uses adaptive clustering to provide a provable guarantee of k-anonymity in data streams [8], [14]. In CASTLE, tuples from a data stream are accumulated to form dynamic generalisations over similar QI values, which are output after time δ as a k-anonymised cluster. Therefore, a data processor can give a guarantee of k-anonymisation. However, it has been shown that k-anonymised data streams are vulnerable to several well-documented privacy attacks [11], [15]. These privacy vulnerabilities include:

1) **Enumeration**: The number of tuples in an input stream is equivalent to those in the anonymised stream.
2) **Boundary Observation**: Tuples can be reidentified by observing the extreme values disclosed by generalisation boundaries.
3) **Homogeneity**: Tuples in the same generalisation may share a_s values leading to homogeneity.
4) **Duplication**: Tuples from the same user may be used to satisfy k-anonymity, limiting individual privacy.

Vulnerabilities 1-4 mean that k-anonymity is a weak privacy guarantee in practice [11]. The use of distinct *pid* values to determine generalisation size means that vulnerability 4 does not apply to data streams anonymised by CASTLE but the existence of vulnerabilities 1-3 motivate the exploration of stronger privacy guarantees than k-anonymity for data streams.

C. l-diversity in Data Streams

We model a data stream as containing a sensitive attribute field a_s. This is a data field containing sensitive information that is required by a data processor. This field is not subject to manipulation and is therefore not generalised. l-diversity is a strengthened form of k-anonymity that is used to protect sensitive attribute a_s from homogeneity attacks [15]. This is achieved by enforcing a lower bound on the distinct values of a_s for a generalisation:

Definition (l**-diverse Data Streams**). *Consider a data stream $S(pid,\ QI,\ a_s)$. An anonymised data stream S' satisfies l-diversity over QI and a_s if and only if every generalisation of QI has at least l distinct values of a_s.*

Achieving l-diversity ensures that homogeneity attacks can no longer be performed on S'. This is because the requirement for diversity is enforced on a_s.

D. Differential Privacy in Data Streams

Differential privacy is a strong privacy constraint that can be applied to data streams [9], [16]:

Definition ((ϵ, δ) **Differentially Private Data Streams**). *Let A be a randomised algorithm that takes a data stream S as input and outputs some $S' \in O$, where O is the image of A. A satisfies $(\epsilon,\ \delta)$ differential privacy if, for all outputs $S' \in O$ and all input data streams S, S_{-t} which vary by a single tuple t, the following holds with probability δ:*

$$e^{-\epsilon} \leq \frac{Pr[A(S) = S']}{Pr[A(S_{-t}) = S']} \leq e^{\epsilon} \tag{1}$$

Differential privacy can be applied under two models, these being interactive and non-interactive differential privacy [12]. Under an interactive model, a data controller provides some interface which data processors can use to execute queries on a data stream. Under a non-interactive model, a data controller instead gives data processors full access to a differentially private version of a data stream. Interactive models are more effective in restricting an adversary's knowledge but suffer under repeated observation [17].

PeGaSus is a prominent algorithm that applies interactive differential privacy to data streams through the use of a Perturber, Grouper and Smoother [16]. Although PeGaSus operates with low error compared to other differentially private data streams, its reliance on an interactive model makes it an unsuitable for a problem context where the data controller and data processors are separate entities. Data controllers are likely to prefer query agnostic data publishing mechanisms and data processors may dislike constraints on processing set by an interactive model. This context, where data controllers wish to publish a data stream to third-party data processors is the focus of CASTLE and CASTLEGUARD.

P^2RoCAL [18] is an alternative method of anonymising data streams which translates a stream of sensitive data into a synthetic data stream with a similar statistical distribution using data condensation and rotational perturbation. This method is considered a very strong solution which provides efficient, strong privacy, however, it does not provide data subjects with a formalised privacy guarantee in the same way as an algorithm satisfying k-anonymity or (ϵ, δ) differential privacy.

III. Models

This section details the adopted system, fault and data models used in the design and evaluation of CASTLEGUARD.

A. System Model

As in Section II-B, CASTLE is a data stream anonymisation algorithm with the following specification:

> **Input:** A data stream $S(\text{pid}, \text{QI}, a_s)$ and parameters
> **Output:** A k-anonymised data stream $S'(G, a_s)$ where G is a generalisation over QI
> **Safety:** No subset $s' \subseteq S'$ can be used to harm the privacy of any tuple $t \in S$ with reasonable confidence
> **Liveness:** A tuple which enters at time t_1 is either suppressed or output in a cluster by time $t_1 + \delta$, where δ is the delay constraint parameter

We seek to increase the privacy afforded by this model by satisfying l-diversity and differential privacy in addition to k-anonymity. As such, we consider a stronger model:

> **Input:** A data stream $S(\text{pid}, \text{QI}, a_s)$ and parameters
> **Output:** A k-anonymised, l-diverse and differentially private data stream $S'(G, a_s)$
> **Safety:** No subset $s' \subseteq S'$ can be used to harm the privacy of any tuple $t \in S$ with reasonable confidence
> **Liveness:** A tuple which enters at time t_1 is either suppressed or output in a cluster by time $t_1 + \delta$

B. Adversary Model

We evaluate the performance of anonymisation algorithms under an adversary model where:

- An adversary has knowledge of the algorithm used, S', $S \setminus \{t_?\}$ and $t_?$. $t_?$ is a tuple that may or may not be an element of the unanonymised data stream S.
- The adversary wishes to learn whether $t_? \in S$

We can show that the original CASTLE algorithm violates safety under this adversary model.

Proof. Using vulnerability (1) as defined in Section II-B, the adversary computes $C(S \setminus \{t_?\})$ and compares their anonymised data stream with S'.

If $|C(S \setminus \{t_?\})| = |S'|$, then $t_? \notin S$.

Alternatively, if $|C(S \setminus \{t_?\})| < |S'|$, then $t_? \in S$.

In both cases, the privacy of $t_?$ is compromised, resulting in a violation of safety. □

C. Data Model

When considering our input data stream $S(\text{pid}, \text{QI}, a_s)$, we make the assumption that a_s cannot be used by an adversary for the purposes of reidentification. For example, it may be a label used for machine learning; this being potentially sensitive but not sufficient to reidentify a tuple. Anonymisation techniques which do not use this assumption are considered in Section VII-A. For the purposes of analysing performance under an adversary, we assume that the input data stream S is finite and eventually halts. Without this assumption, the decision problem $t \in S'?$ becomes undecidable, thereby limiting the expressiveness of the adversary model.

IV. CASTLEGUARD

In this section we formalise the modifications to CASTLE that are required to satisfy non-interactive differential privacy (Section IV-A) and l-diversity (Section IV-B). We then show that it adheres to the defined system model (Section IV-C) and evaluate it under the adversary model (Section IV-D).

A. Differentially Private k-anonymity

The (k, β)-SDGS algorithm for k-anonymisation with non-interactive differential privacy is a starting point for the incorporation of differentially private k-anonymity [11].

Definition ((k, β)-SDGS). *The (k, β)-SDGS algorithm operates in three stages for an arbitrary input dataset D:*

1) ***β-Sampling**: All tuples in D are sampled with probability β or discarded with probability $1 - \beta$*
2) ***Data-independent Generalisation**: Sampled tuples are grouped using generalised clusters which are determined independently of the input dataset D*
3) ***k-Suppression**: Generalisations are suppressed (not published) if they appear less than k times.*

However, (k, β)-SDGS is not suitable for direct application in data streams as it relies on *a priori* knowledge to perform its data-independent generalisations.

Example. Consider $S(\text{pid}, \{\text{Age}, \text{Salary}\}, a_s)$. We use a Data-independent Generalisation which clusters numerical data into groups $[0 - 99, ...]$. Most Age values are grouped in the $0 - 99$ cluster, resulting in high information loss. Most Salary values are grouped in distinctive clusters (e.g. $24000 - 24099$), reducing privacy by generalisation.
Data-independent Generalisation is one method of achieving safe k-anonymisation, where it becomes impossible to determine tuple values from their generalisations and exploit vulnerability 2 [11]. We consider an alternative method to

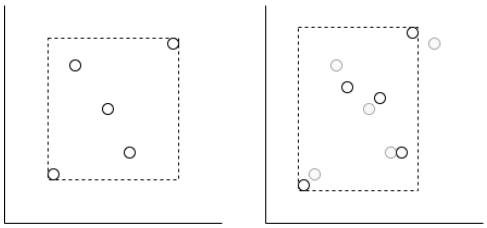

Fig. 1: Example effect of QI perturbation on two dimensions

achieve safe k-anonymisation using perturbation. By overlaying additive noise to a tuple's QI values before clustering, we ensure that an observer cannot determine with certainty any original QI value from the extreme values of a cluster.

Example. An adversary receives a generalisation $[20.2, 32.6]$ over an Age attribute from S'. They cannot determine with certainty that the values 20.2 and 32.6 were present in S due to random perturbation. As such, these values cannot be used to compromise the privacy of tuples in S', but continue to provide utility to a data processor.

By applying perturbation to the β-sampled tuples yielded by the data stream and enforcing k-Suppression in publication, the definition of differentially private k-anonymity can be adapted to apply to data streams.

Definition (Differentially Private k-anonymity in Data Streams). *Algorithm A satisfies differentially private k-anonymity over an arbitrary input data stream D if:*

> *β-**Sampling**: When D yields a tuple it is immediately suppressed with probability $1 - \beta$*
> ***Perturbation**: Sampled tuples have their QI values perturbed using additive noise and are grouped using generalised clusters over perturbed values.*
> ***k-Suppression**: Generalisations are suppressed (not published) if they appear less than k times.*

If a constant scale of noise is used to perform perturbation in a data stream, attributes with large ranges will receive insufficient perturbation and attributes with small ranges will receive too much perturbation. Therefore, additive noise must scale independently according to each QI attribute's global range. The range does not have to be known *a priori*, since it can be dynamically updated using values encountered.

We implement perturbation using the Laplace mechanism [9] with $\mu = 0,\ b = \frac{R}{\phi}$, where R is the encountered global range of a QI attribute and ϕ is a privacy parameter. R can be considered the worst-case sensitivity of a range query on a QI attribute. We denote the privacy parameter as ϕ rather than ϵ because the algorithm is not ϕ-differentially private; the full algorithm also publishes the sensitive attribute without the Laplace mechanism. We can use ϕ to control the probability that any QI value will be perturbed by less than $r \cdot R$, $r < 0$.

Theorem 1. *Given a Laplacian distribution used for perturbation with mean 0 and scale $\frac{R}{\phi}$ and an unperturbed QI value v with perturbed value v', for any QI attribute with global range R and any proportion $r > 0$:*

$$Pr\left[|v - v'| \ < r \cdot R\right] = 1 - e^{-r \cdot \phi} \tag{2}$$

Algorithm 1: CASTLEGUARD Algorithm

input: Data stream S with schema $S(\text{pid}, \text{QI}, a_s)$
input: Integer $k > 0$ for k-anonymity
input: Integer $l > 0$ for l-diversity
input: Real number $0 < \beta < 1$ for β-sampling
input: Real number $\phi > 0$ for perturbation
input: Integer $\delta > 0$ for the delay constraint
input: Integer $b > 0$ for maximum active clusters
out : Data stream S' with schema $S'(G, a_s)$

```
while S is not empty do
    Let t be the next tuple from S;
    if random(0, 1) > β then
        Suppress t;
    else
        Let t' ← perturb(t);
        Let C ← best_selection(t');
        if C is null then
            Create a new cluster on t';
        else
            Insert t' into C;
        end
    end
    Let t_δ be the tuple at position t'.p − δ;
    delay_constraint(t_δ);
end
```

Proof. *Given a noise value x sampled from a Laplace distribution with mean 0 and scale $\frac{R}{\phi}$, for any $r > 0$:*

$$\begin{aligned} Pr\left[-r \cdot R \ < \ x \ < \ r \cdot R\right] &= CDF\left(r \cdot R\right) - CDF\left(-r \cdot R\right) \\ &= 2 \cdot \left(CDF\left(r \cdot R\right) - CDF(\mu)\right) \\ &= 2 \cdot \left(1 - \frac{1}{2} \cdot e^{-\frac{r \cdot R - 0}{\left(\frac{R}{\phi}\right)}} - \frac{1}{2}\right) \\ &= 1 - e^{-\frac{r \cdot R}{\left(\frac{R}{\phi}\right)}} \\ &= 1 - e^{-r \cdot \phi} \end{aligned}$$

Thus, for any unperturbed QI value v with perturbed value $v' = v + x$ where x is again a noise value sampled from the Laplace distribution, we find that:

$$Pr\left[|v - v'| \ < \ r \cdot R\right] = 1 - e^{-r \cdot \phi} \tag{3}$$

We can express ϕ in terms of a probability and a limit r:

$$\phi = -\frac{ln(1 - Pr\left[|v - v'| \ < \ r \cdot R\right])}{r} \tag{4}$$

□

Following from the definition of differentially private k-anonymity for data streams, we implement these properties by adding parameters β and ϕ and making modifications to its procedures [8]. We refer to this revised form of the algorithm as CASTLEGUARD, for *Continuously Anonymising STreaming data via adaptive cLustEring with GUARanteed Differential privacy*. Henceforth β refers to the probability used for β-sampling and b as the maximum size of Γ. The CASTLEGUARD procedure is represented in Algorithm 1.

Algorithm 2: $perturb(t)$

input: Tuple t
out : Perturbed tuple t'

1 **for** $A \leftarrow QI$ **do**
2 A.min $\leftarrow min$(A.min, t.A);
3 A.max $\leftarrow max$(A.max, t.A);
4 t'.A $\leftarrow t$.A + Laplace($\mu = 0$, $b = \frac{A.max - A.min}{\phi}$)
5 **end**
6 **return** t'

Lines 3-4 of Algorithm 1 show an implementation of β-sampling. Line 6 utilises an implementation of the perturbation algorithm described in Algorithm 2. An implementation of k-suppression is not explicitly required as CASTLE already satisfies k-anonymity. $best\_selection$ identifies the cluster with minimum enlargement following the insertion of a tuple and $delay\_constraint$ releases or suppresses a tuple if its cluster satisfies k-anonymity; both remain unchanged [8].

These changes enforce that all clusters are generalised using differentially private k-anonymity. Therefore, we can conclude that CASTLEGUARD as a whole satisfies differentially private k-anonymity. Further, we can guarantee the level of differential privacy by applying β-sampling [11]:

Theorem 2. *A differentially private k-anonymous algorithm with β-sampling satisfies (ϵ, δ) differential privacy for any $\epsilon \geq -ln(1-\beta)$ and $\delta = d(k, \beta, \epsilon)$:*

$$d(k, \beta, \epsilon) = \max_{n : n \geq \lceil \frac{k}{\gamma} - 1 \rceil} \sum_{j > \gamma n}^{n} f(j; n, \beta) \tag{5}$$

$$f(j; n, \beta) = \textit{PMF of a Binomial Distribution} \tag{6}$$

$$\gamma = \frac{(e^{\epsilon} - 1 + \beta)}{e^{\epsilon}} \tag{7}$$

A derivation of Theorem 2 can also be found in [11]. In summary, an extension of differentially private k-anonymity is proposed for data streams using perturbation. This property is then applied to the CASTLE algorithm to achieve a privacy guarantee of (ϵ, δ) differential privacy in addition to k-anonymity. With reference to the problem context, this guarantee supports the goal of data controllers to provide a strong, provable guarantee of privacy to data subjects.

B. l-diversity

We implement l-diversity by applying modifications to the cluster operation of the CASTLE algorithm [8]. These changes enforce that all clusters output by CASTLE satisfy l-diversity and that clusters which do not satisfy l-diversity are not output and are suppressed. We introduce a parameter l to control the l-diversity of S' with respect to a_s. This allows us to add an additional guarantee of l-diversity to the output data stream in addition to k-anonymity and differential privacy.

C. Adherence to System Model

We now demonstrate that CASTLEGUARD adheres to the defined system model following the incorporation of new algorithms to provide strong privacy guarantees.

Input: A data stream $S(\text{pid}, \text{QI}, a_s)$ and parameters. The input data stream schema is unchanged as no additional attributes or *a priori* information is required. New parameters β, ϕ and l are introduced.
Output: A k-anonymised, l-diverse and differentially private data stream $S'(G, a_s)$. Adherence to l-diversity and differential privacy is demonstrated in Sections IV-A and IV-B. The output data stream schema is unchanged.
Safety: No subset $s' \subseteq S'$ can be used to harm the privacy of any tuple $t \in S$ with reasonable confidence. This is demonstrated in Section IV-D.
Liveness: A tuple which enters at time t_1 is either suppressed or output in a cluster by time $t_1 + \delta$. We do not change CASTLE's $delay\_constraint$ method, which is used to control the output of tuples, and therefore maintain this liveness property. However, we do provide more ways that a tuple may be suppressed, either through β-sampling or l-diversity suppression.

D. Performance Under Adversary Model

We now consider the performance of CASTLEGUARD under the adversary model defined in Section III. We prove that its safety property is maintained even under a knowledgeable adversary, such that the privacy of an individual tuple cannot be compromised with reasonable confidence.

Proof. An adversary cannot learn anything with certainty from comparing $|CG(S \setminus \{t_?\})|$ and $|S'|$ due to β-sampling, even if they know β, because the sampled data stream is likely to be different in each execution. Let MaxN : $(S \cup \{t_?\}) \times S' \times QI \rightarrow \mathbb{R}$ be a relation which defines, for a generalisation $G \in S'$, an unperturbed tuple $t \in S \cup \{t_?\}$ where t falls within the bounds of G and a QI attribute a, the maximum absolute magnitude of noise max that t could receive on attribute a and remain within the bounds of G. From Theorem 1, we have (8).

$$\Pr\left[|x| \ \leq \max\right] = 1 - e^{-\frac{\max}{R_a} \cdot \phi} \tag{8}$$

The probability of t generalising to G, written $t \rightarrow G$, after perturbation is given by (9).

$$\Pr[t \rightarrow G] = \beta \cdot \prod_{a \in \text{QI}} \left(1 - e^{-\frac{\max}{R_a} \cdot \phi}\right) \tag{9}$$

Let IntN : $(S \cup \{t_?\}) \times S' \times QI \rightarrow \mathbb{R}^2$ be a relation which defines, for a generalisation $G \in S'$, an unperturbed tuple $t \in S \cup \{t_?\}$ where t does not fall within the bounds of G and a QI attribute a, the interval of possible noise values $[min, max]$ on t which would allow it to fall within the bounds of G. From Theorem 1, we derive (10).

$$\Pr\left[x \in \text{IntN}(t, G, a)\right] = e^{-\frac{\min}{R_a} \cdot \phi} - e^{-\frac{\max}{R_a} \cdot \phi} \tag{10}$$

Hence the probability of $t \rightarrow G$ after perturbation is (11).

$$\Pr[t \rightarrow G] = \beta \cdot \prod_{a \in \text{QI}} \left(e^{-\frac{\min}{R_a} \cdot \phi} - e^{-\frac{\max}{R_a} \cdot \phi}\right) \tag{11}$$

Consider any generalisation $G \in S'$ sharing an a_s value with $t_?$. If $t_?$ falls within G, then (9) is the probability

of $t_?$ remaining in G after anonymisation and (11) is the probability of a tuple $t \neq t_?, t.a_s = t_?.a_s$ being perturbed into G after anonymisation. Otherwise, if $t_?$ does not fall within G, then (11) is the probability of $t_?$ being perturbed into G after anonymisation and (9) is the probability of a tuple $t \neq t_?, t.a_s = t_?.a_s$ remaining in G after anonymisation.

For $\beta < 1$, $\phi > 0$ and $max \neq min$, (9) and (11) are never certain. As such, an observer cannot know with confidence whether any generalisation $G \in S'$ was derived from $t_?$ or a different tuple t which shares an a_s value with $t_?$.

Considering the negative case, where no generalisations in S' share an a_s value with $t_?$, the adversary still cannot know whether $t_?$ was not present in S or whether it was removed via β-sampling given that they do not know $\Pr[t_? \in S]$:

$$\Pr[t_?[a_s] \notin S'] = \Pr[t_? \notin S] + (1 - \beta)(\Pr[t_? \in S]) \quad (12)$$

We conclude that an adversary cannot learn whether $t_? \in S$ with certainty under this model, protecting the privacy of $t_?$ and maintaining safety under a knowledgeable adversary. □

V. Experimental Setup

In this section we detail the experimental setup used to analyse CASTLEGUARD. This includes details of the data sets and measures used to produce the results presented in Section VI.

CASTLEGUARD was implemented in Python [19]. In order to extend the implementation to support the anonymisation of categorical data, as is possible in CASTLE, we define a relation that maps leaf-level elements of the Domain Generalisation Hierarchy (DGH) to numerical values dynamically, thereby allowing the system to perform perturbation. If desirable, these numerical values can be translated back to categorical elements once published. The experiments investigate the effect of CASTLEGUARD parameters on its performance, measured in terms of the data quality and information loss with regard to a CASTLEGUARD anonymised data stream. The non-determinism associated with the use of differential privacy motivated five repeated experiments, with all measurements showing negligible variance between these experiments.

A. Information Loss

Information loss was measured using a sample of 1000 records from the January 2019 Yellow Taxi Trip Records from the New York City Taxi and Limousine Commission [20]. We configure *TLC-Taxis* by modifying *VendorID* to simulate a unique identifier, *PickupLocationID* and *TripDistance* to be quasi-identifiable attributes, and *FareAmount* to be the protected attribute. The average information loss is calculated as a process of the CASTLEGUARD algorithm and is reported after anonymising the sampled data stream in full. The effects of varying b and μ are recorded to measure how far CASTLEGUARD adheres to the same information loss trends as CASTLE [8]. The effects of varying β and ϕ are recorded to measure whether the introduction of differentially private k-anonymity affects the information loss of the algorithm.

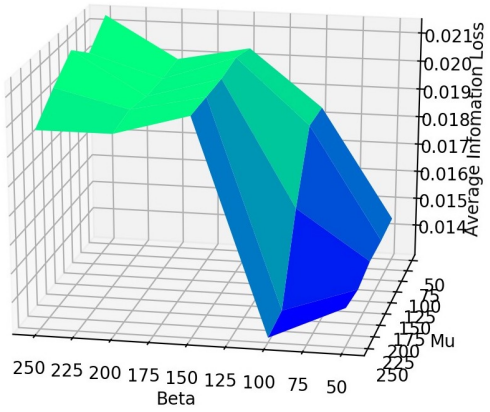

Fig. 2: Information Loss on b and μ (*New York Taxi [20]*, k = 10, l = 1, δ = 200, β = 1 and ϕ = 10000)

B. Quality Metrics

Data quality was measured using the Pima Indians Diabetes data set of the UCI Machine Learning training collection [21]. We configure the feature attributes of the data set to be quasi-identifiable attributes, with the label *outcome* as the protected attribute. Data quality is calculated by comparing the test error of k-Nearest Neighbours (k-NN) and Neural Network (NN) machine learning classifiers after training on both the unanonymised data stream and the CASTLEGUARD anonymised data stream. Both classifiers are trained by considering the minimum and maximum values of each QI attribute as separate features. The effects of varying β and ϕ are recorded to measure the impact of increasing sampling and perturbation on the data quality.

VI. Results

In this section we present the results of experimentation on CASTLEGUARD. As in Section V, these result focus on the measurement of information loss and data quality.

A. Information Loss - The Effects of b and μ

Parameter b controls the number of active clusters, whereas μ is the number of most recently generalised clusters on which the average information loss is calculated. Fig. 2 shows the average information loss of anonymised tuples over b and μ. The average information loss against b and μ for CASTLEGUARD is commensurate with that of CASTLE [8]. On this new data set, the experiments show a peak of information loss at around b=200 with adjacent troughs. The parameter μ has a consistent effect, increasing with information loss for small b values. As such, CASTLEGUARD exhibits information loss trends that are commensurate with CASTLE.

B. Information Loss - The Effects of β and ϕ

We find that ϕ and β have a small effect on the information loss and may be considered independent from it, with the range of collected values being between 0.0178-0.0182 (*New York Taxi [20]*, k = 10, l = 1, δ = 200, μ = 100, b = 25). This

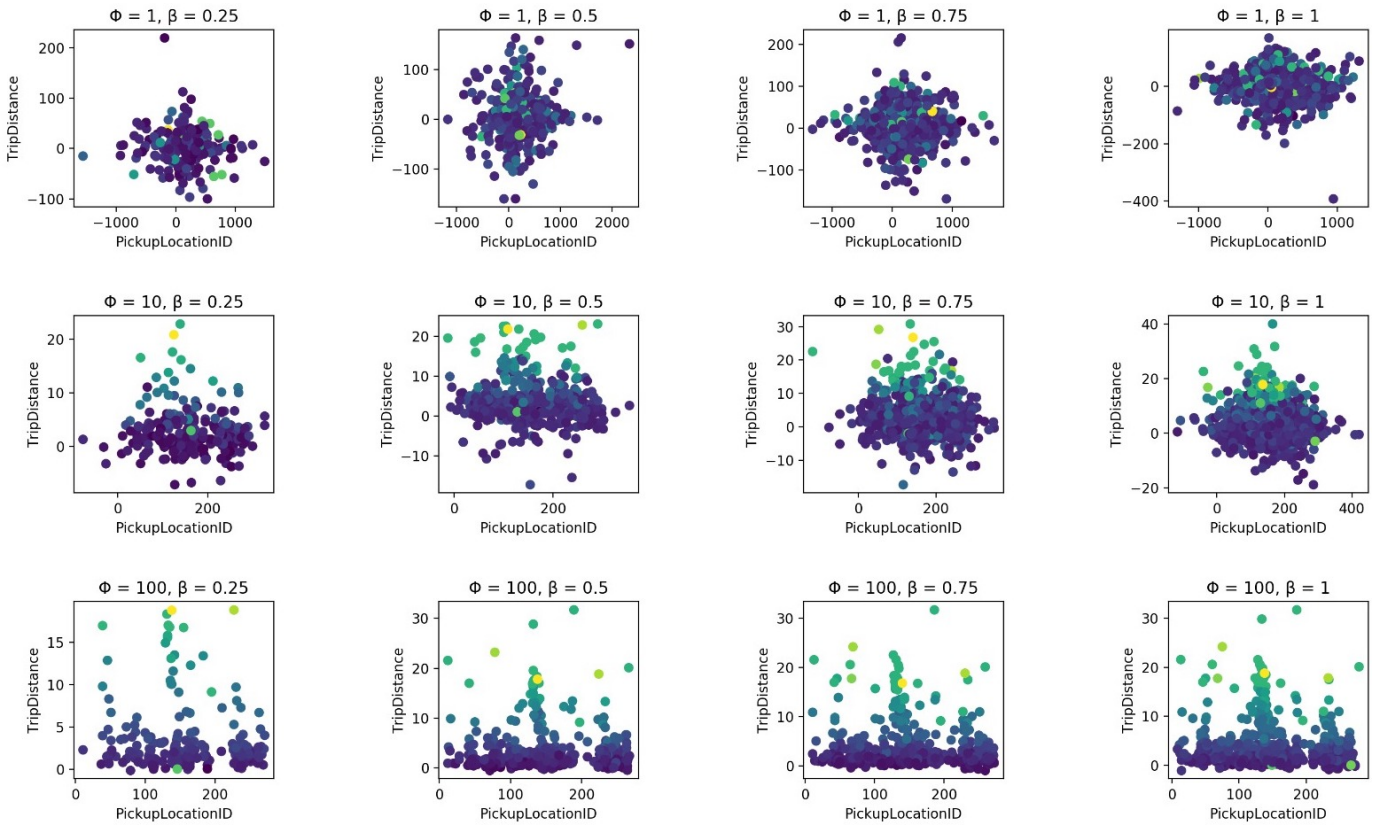

Fig. 3: Perturbed and sampled tuples in CASTLEGUARD on β, ϕ (*New York Taxi [20]*). Tuple colour represents *FareAmount*. With low values of ϕ and β (top and left), the shape of the distribution is noisy and sparse. With higher values of ϕ and β (bottom and right), the shape of the distribution becomes more similar to that of the input distribution.

demonstrates that the inclusion of differential privacy does not affect information loss. Instead, ϕ can be considered to control the quality of the output stream and β considered to control the quantity of information in the output stream. This effect can be visualised by comparing the distribution of sampled and perturbed input tuples on the dataset, illustrated in Fig. 3.

C. Data Quality

As illustrated in Fig. 4, the performance of using k-NN on CASTLEGUARD data is generally equivalent, if not better, than training on original data (54.87%). A maximum accuracy of 65.9% was achieved (ϕ = 100, β = 0.25), indicating a level of data quality that can sustain many applications. Performance in accuracy is maximised for medium or large ϕ and medium or low β. We conjecture this model performs well because both k-NN and CASTLEGUARD use similarity-based clustering.

Fig. 5 considers data quality after training on a NN. In this case, accuracy on the anonymised data stream is consistently below the control accuracy (70.56%), reaching a maximum of 68.4% (ϕ = 100, β = 0.5). Performance in Fig. 5 does not appear to clearly correlate with ϕ and β. However, using AUC as a measurement for classification performance Fig. 6 establishes a relationship between ϕ and the model's AUC-ROC. This demonstrates that the model is better at distinguishing between classes with higher ϕ values. We conclude a positive correlation between ϕ and both classifier accuracy

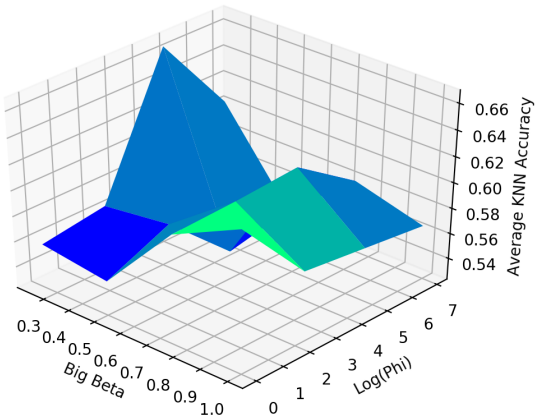

Fig. 4: k-NN accuracy on β, ϕ (*Pima Indians*, k = 7, l = 1, δ = 100, μ = 100, b = 25)

and AUC-ROC, supporting our claim that a larger ϕ increases information quality. The k-NN model was the most performant with regard to accuracy. A ϕ of 100 and β between 0.25-0.75 generally resulted in better performance across all models.

These results demonstrate CASTLEGUARD is capable of providing provably anonymised data streams with utility to data processors. Data controllers can therefore provide privacy to data subjects without unduly limiting the capabilities of data processors. Like CASTLE, data quality following anonymisa-

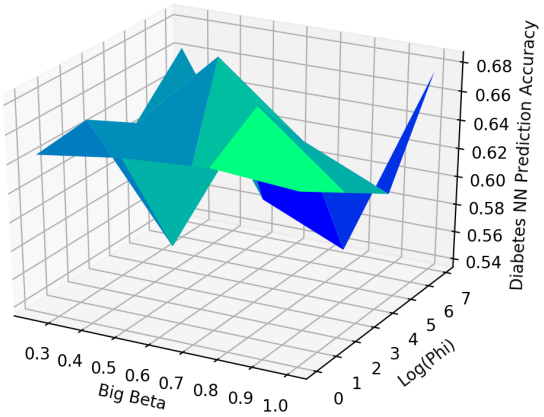

Fig. 5: NN accuracy on β, ϕ (*Pima Indians*, k = 7, l = 1, δ = 100, μ = 100, b = 25)

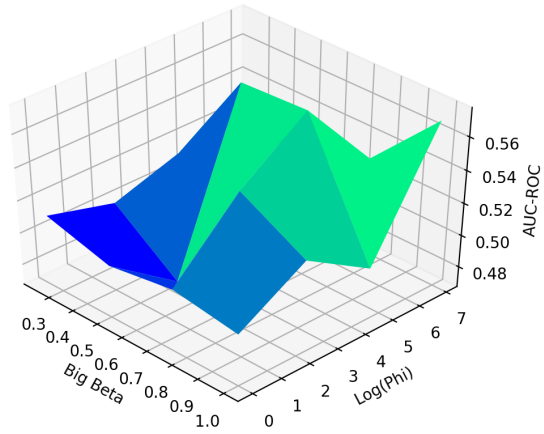

Fig. 6: AUC-ROC of NN on β, ϕ (*Pima Indians*, k = 7, l = 1, δ = 100, μ = 100, b = 25)

tion appears to be domain-dependent, though k-NN models did outperform NN models for accuracy.

VII. Conclusion

This paper presented CASTLEGUARD, a novel data stream anonymisation system that provides guaranteed k-anonymity, l-diversity and differential privacy. Building on CASTLE [8], it was shown that, with safe k-anonymisation and β-sampling, the system satisfies differentially private k-anonymity. The efficacy of the system was demonstrated in the context of machine learning, with experiments showing that it can protect user privacy whilst maintaining data stream utility.

A. Future Work

As an extension of l-diversity, implementing t-closeness would improve privacy by enforcing a representative distribution of values for each generalisation [22]. Also, introducing additive noise perturbation for the sensitive attribute a_s would increase privacy for contexts where a_s is also a quasi-identifiable attribute. A substantial ambition would be the translation of CASTLEGUARD to a local differential privacy model, allowing users to contribute private data to a stream without the need for trust to be placed in a central authority [23].

References

[1] P. Voigt and A. von dem Bussche, *The EU General Data Protection Regulation: A Practical Guide*, 1st ed. Springer, August 2017.

[2] L. Qiu, Y. Li, and X. Wu, "Protecting business intelligence and customer privacy while outsourcing data mining tasks," *Knowledge and Information Systems*, vol. 17, pp. 99–120, November 2008.

[3] M. M. Merener, "Theoretical results on de-anonymization via linkage attacks," *Transactions on Data Privacy*, vol. 5, no. 2, pp. 377–402, August 2012.

[4] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proceedings of the IEEE Symposium on Security and Privacy*. California, USA: IEEE, May 2008, pp. 111–125.

[5] IMEC-DistriNet (KU Leuven), "Linddun official website," 2017. [Online]. Available: linddun.org

[6] C. Systems, "Consumer Privacy Survey: The growing imperative of getting data privacy right," *CISCO CYBERSECURITY*, 11 2019.

[7] 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, May 2002.

[8] J. Cao, B. Carminati, E. Ferrari, and K.-L. Tan, "CASTLE: Continuously anonymizing data streams," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 3, pp. 337–352, January 2011.

[9] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*, vol. 3876. New York, USA: Springer, March 2006, pp. 265–284.

[10] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, August 2014.

[11] N. Li, W. Qardaji, and D. Su, "On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy," in *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*. Seoul, Korea: ACM, May 2012, pp. 32–33.

[12] D. Leoni, "Non-interactive differential privacy: a survey," in *Proceedings of the First International Workshop on Open Data*, 2012, pp. 40–52.

[13] P. Coetzee, M. Leeke, and S. Jarvis, "Towards unified secure on- and off-line analytics at scale," *Parallel Computing*, vol. 40, no. 10, pp. 738–753, December 2014.

[14] P. Wang, J. Lu, L. Zhao, and J. Yang, "B-castle: An efficient publishing algorithm for k-anonymizing data streams," in *Proceedings of the 2nd WRI Global Congress on Intelligent Systems*, vol. 2. Wuhan, China: IEEE, December 2010, pp. 132–136.

[15] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-Diversity: Privacy beyond k-Anonymity," *ACM Transactions on Knowledge Discovery From Data*, vol. 1, no. 1, pp. 3–es, March 2007.

[16] Y. Chen, A. Machanavajjhala, M. Hay, and G. Miklau, "Pegasus: Data-adaptive differentially private stream processing," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. Texas, USA: ACM, October 2017, pp. 1375–1388.

[17] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *Proceedings of the 42nd ACM Symposium on Theory of Computing*. Massachusetts, USA: ACM, June 2010, pp. 715–724.

[18] M. Chamikara, P. Bertok, D. Liu, S. Camtepe, and I. Khalil, "Efficient data perturbation for privacy preserving and accurate data stream mining," *Pervasive and Mobile Computing*, vol. 48, pp. 1–19, August 2018.

[19] A. Robinson, F. Brown, N. Hall, A. Jackson, and G. Kemp, "CASTLEGUARD," May 2020. [Online]. Available: github.com/hallnath1/CASTLEGUARD

[20] City of New York, "2018 yellow taxi trip data," data.cityofnewyork.us/Transportation/2018-Yellow-Taxi-Trip-Data/t29m-gskq, 2018.

[21] UCI Machine Learning, "Pima indians diabetes database," kaggle.com/uciml/pima-indians-diabetes-database, 2016.

[22] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *Proceedings of the 23rd IEEE International Conference on Data Engineering*. Istanbul, Turkey: IEEE, April 2007, pp. 106–115.

[23] G. Cormode, S. Jha, T. Kulkarni, N. Li, D. Srivastava, and T. Wang, "Privacy at scale: Local differential privacy in practice," in *Proceedings of the 2018 International Conference on Management of Data*. Texas, USA: ACM, May 2018, pp. 1655–1658.