RESEARCH

Pseudonymization Risk Analysis in Distributed Systems

Geoffrey K Neumann[1], Paul Grace[2*], Daniel Burns[3] and Mike Surridge[4]
gkn@it-innovation.soton.ac.uk, pjgrace@gmail.com, dkb@it-innovation.soton.ac.uk, ms@it-innovation.soton.ac.uk

[*]Correspondence: pjgrace@gmail.com
[2]IT Innovation, University of Southampton, Gamma House, Enterprise Road, SO16 7NS Southampton, UK
Full list of author information is available at the end of the article

Abstract

In an era of big data, online services are becoming increasingly data-centric; they collect, process, analyze and anonymously disclose growing amounts of personal data in the form of pseudonymized data sets. It is crucial that such systems are engineered to both protect individual user (data subject) privacy and give back control of personal data to the user. In terms of pseudonymized data this means that unwanted individuals should not be able to deduce sensitive information about the user. However, the plethora of pseudonymization algorithms and tuneable parameters that currently exist make it difficult for a non expert developer (data controller) to understand and realise strong privacy guarantees. In this paper we propose a principled Model-Driven Engineering (MDE) framework to model data services in terms of their pseudonymization strategies and identify the risks to breaches of user privacy. A developer can explore alternative pseudonymization strategies to determine the effectiveness of their pseudonymization strategy in terms of quantifiable metrics: i) violations of privacy requirements for every user in the current data set; ii) the trade-off between conforming to these requirements and the usefulness of the data for its intended purposes. We demonstrate through an experimental evaluation that the information provided by the framework is useful, particularly in complex situations where privacy requirements are different for different users, and can inform decisions to optimize a chosen strategy in comparison to applying an off-the-shelf algorithm.

Keywords: privacy; pseudonymization; risk analysis

1 Introduction

Motivation. The creation of novel, personalized and optimized data-centered applications and services now typically requires the collection, analysis and disclosure of increasing amounts of data. Such systems will leverage data-sets that include personal data and therefore their usage and disclosure represent a risk to a user's (data subject) privacy. This data may be disclosed to trusted parties or released into the public domain for social good (e.g. scientific and medical research); it may also be used internally within systems to improve the provision of a service (e.g. to better manage resources, or optimize a service for individual users). Importantly, users have different viewpoints of what privacy means to them. Westin's privacy indexes provide evidence of this [1]. For example, users may wish to disclose sensitive information to support medical research, but not allow an environmental monitoring service to track their movements. However, there is a concern that individuals

may simply not understand the implications for their own data within a service [2]; or this may reflect the complexity of technology itself [3]. Hence, systems should be developed to ensure that: i) each individual's privacy preferences are taken into account, and ii) risks to each and every user's privacy are minimized.

Data *anonymization* is the process of removing directly identifying information from data. Anonymized datasets may contain both sensitive information, and information that could identify the person. Hence, concerns about preserving privacy has led to algorithms to pseudonymize data in such a way that personal information is not disclosed to an unwanted party, e.g. k-anonymization [4], l-diversity [5] and t-closeness [6]. In general, developers' understanding of external privacy threats are limited [7]. Non-experts in these algorithms will also find them difficult to understand in terms of: i) the impact of the risk of re-identification, and ii) the effect of the pseudonymization on the usefulness of the data for its intended purpose. Hence, development frameworks that guide a privacy-by-design [8] process to achieve better privacy protection are required.

Contribution. In this paper we propose a Model-Driven Engineering (MDE) framework to model and analyze the privacy risks of a system employing pseudonymization. This provides the following key features:

- *Modeling user privacy.* The framework takes as input models of the system in terms of the data and its flow between systems and stakeholders (via data flow diagrams) and automatically generates a formal model of user privacy in the system (in terms of a Labeled Transition System).
- *Pseudonymization risk analysis.* The user-centered privacy model is then analyzed to automatically identify and quantify privacy risks based upon the modeled choices of pseudonymization. This returns the number of privacy violations that a pseudonymization strategy will or may result in, and an assessment of whether the data will still be useable for its intended purposes when the violations have been eliminated to an acceptable standard.

We argue that the framework can be used by non-pseudonymization experts at different stages of the development lifecycle to make informed decisions about: pseudonymization strategies, whether pseudonymized data is safe to release, and whether to impose restrictions on when data should be released. A system designer, using profiled user data, can understand privacy risks early in the design process, and ensure that user privacy is a foundational requirement to be maintained. The framework can also be used by systems developers (or data controllers) to make decisions about data disclosure based on the information from the users in the real dataset.

Outline. Section 2 first describes how the framework is used to model user-centred privacy-aware systems that leverage pseudonymization, before describing how potential pseudonymization risks are identified in Section 3. Section 4 reports how the framework carries out data utility analysis. Section 5 describes how the framework is applied following privacy-by-design practice, and the framework is evaluated in Section 6. Section 7 describes related work, and Section 8 provides a conclusion and indicates areas of future work.

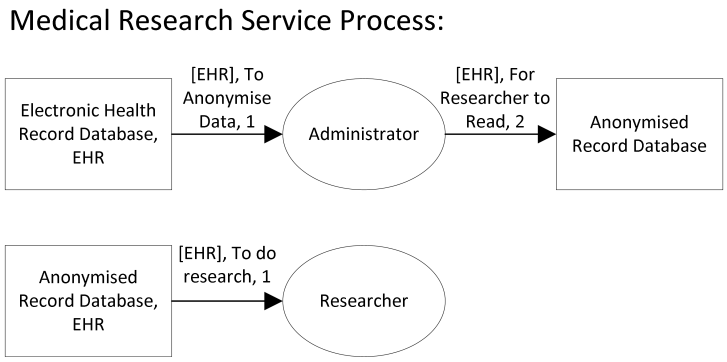

Figure 1 Example data-flow diagram for sharing on data

2 Modelling Privacy Aware Pseudonymization Systems

Here we describe a framework to model privacy aware systems. This follows two steps. First, the developer models their system; second, a formal model of user privacy in this system is generated.

2.1 Step 1: modelling a privacy aware system

The developers of the system create a set of artifacts that model the behaviour of their system:

- A set of *Data-Flow diagrams* that model the flow of personal data within a system. In particular focusing on how data is exchanged between the actors and datastores. We utilize data-flow diagrams because they are an existing well-understood form of modeling that simply captures data behaviours.
- The *data schema* associated with each datastore; this highlights the individual fields and also those containing personal data, and potentially sensitive personal data.

We now consider a simple example to illustrate how these elements form the input to the modelling framework. Two data-flow diagrams are given in Figure 1. The nodes represent either an `actor` (oval) or a `datastore` (rectangle). The datastores are labelled by two objects: the first is the identifier for the datastore, and the second are the data schemas. The actual data flow is represented by directed arrows between the ovals and rectangles, henceforth referred to as flow arrows. Each flow arrow is labelled with three objects: the set of data fields which flows between the two nodes, the purpose of the flow, and a numeric value indicating the order in which the data flow is executed. We assume datastore interfaces that support querying and display of individual fields (as opposed to coarse-grained records). The example illustrates the process where an administrator queries data from an healthcare database to produce a pseudonymized version that is stored in a data store; from which data is queried by researchers.

2.2 Step 2: automatically generating an LTS privacy model

In this section we provide a formal model of user privacy that is generated based upon the input data-flow diagrams (we do not detail the transformation algorithm in this paper, see [9]). User privacy is modeled in terms of how actor actions on

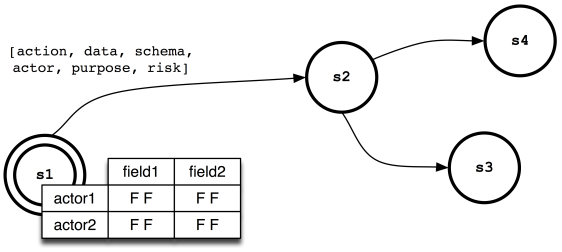

Figure 2 A State-based Model of User Privacy

personal data change the user's state of privacy. We define an `actor` to be an individual or role type which can identify the user's personal data. Depending on the service provided, each actor may or may not have the capability to identify personal data. Hence, a user's privacy changes if any of their personal data has been or can be identified by an actor. Prior models following this approach are: a Finite State Machine (FSM) [10] [11] or a Labelled Transition System (LTS) [12]. The common theme in both is that the user's privacy at any point in time is represented by a state, and that actions, executed by actors, taken on their personal data can change this state. We build upon these approaches and extend them to label both states and transitions in such a way that the model can be analysed to understand how, and why, the user's privacy changes. This novel contribution allows us to represent not only the sharing of a user's personal information, but also the potential for a user's personal information to be shared. This is the case when personal information is stored in a datastore that can be accessed by multiple individuals.

The key elements of our model (illustrated in Figure 2) are:

- *States*: are representations of the user's privacy. They are labelled with variables to represent two pre-dominant factors: whether a particular actor *has* identified a particular field, or whether an actor *could* identify a field. These variables, henceforth known as *state variables*, take the form of Booleans, and there are two for each actor-data field pair (has, could). The state label s1 is given the table values shown in Figure 2.
- *Transitions*: represent `actions` (collect, create, read, disclose, anon, delete) on personal data performed by `actors`. They are labelled according to: i) an `action`, ii) the set of `data fields`, iii) the `data schema` that the data field is a part of, iv) the `actor` performing the action. There are two optional fields: i) a `purpose` that explains the reason a particular privacy action is being taken, and ii) a `privacy risk measure` to identify risks associated with this action (whose value is calculated and annotated during risk analysis).
- *Pseudonymization*: the disclosure of pseudonymized versions of each sensitive field is modelled using the `anon` transition. State variables (i.e. can access, has accessed) can be declared on these fields. For example an analyst may have access permission for the field $weight_{anon}$ but may not have permission to access $weight$. This will mean that they may be allowed access to pseudonymized weight data for statistical purposes but should be prevented from matching any value to an individual.

2.3 Step 3: Considering User Privacy Preferences

Each user has a policy which controls all aspects of how their data is allowed to move through the system. This section will in brief outline how these policies operate. As they control more aspects of the system than just pseudonymization, for a more detailed description refer to literature describing the system overall [12]. These policies, which are derived from information obtained from a questionnaire, have three main aspects:

1. Which services the user agrees to use (based upon the purpose of the service, and optionally their trust of actors in that service).
2. The sensitivities the user has about certain fields, represented by either a sensitivity category (low, medium, high for example), or a number which takes a value between 0 and 1 indicating how sensitive the user is to disclosure of that data.
3. The overall *sensitivity* preferences of a user. Users are categorised as having one of three preference levels: *unconcerned*, *pragmatist* and *fundamentalist* [1]. These three categories indicate, respectively, that the user requires a low, medium or high level of restriction on data and which actions are allowed on their data. An *unconcerned* user will have few restrictions while a *fundamentalist* user will have many restrictions.

Within the context of the LTS, user policy operates on transitions and is used to determine whether a transition should or should not be allowed. A transition that occurs despite not being permitted is referred to as a policy violation. An actor that is not permitted to access a given field will be referred to as an *unauthorised actor* while one who is allowed will be referred to as an *authorised actor*.

3 Risk Analysis

3.1 Identifying Pseudonymization Risk Transitions

In the context of this paper *risk* refers to a danger that a single piece of information about a user is obtained by an entity or individual that is not authorised to have access to that information. In the case of pseudonymization this danger takes the form of a quantifiable probability. Some risks may be acceptable and it is impossible to eliminate all risk. In the case of pseudonymization, risks associated with a sufficiently low probability can be ignored. Which risks may be ignored is determined by a combination of system policy and the preferences of the individual user to which the data refers, their *user policy*.

Pseudonymization is incorporated into the LTS via the `anon` transition. This creates pseudonymized versions of each field. State variables (i.e. can access, has accessed) can be declared on these fields in the same way as for non anonymized fields and are also subject to the same permissions. For example, an analyst may have access permission for the field `weight.anon` but may not have permission to access `weight`. This corresponds to an actor having access to a pseudonymized version of the database. When they are only permitted access to pseudonymized versions of sensitives fields they should be prevented from attaching any sensitive values to specific users.

There are two key types of risk which are considered:

1. *Re-identification*: The risk that a person whose personal data is pseudonymized within a disclosed data set can be re-identified.

2 *Value Prediction*: Risk of a sensitive value being matched to an individual.

Techniques such as k–anonymization [4] prevent re-identification but do not guarantee that there is not still a value prediction risk. For example, suppose that after k-anonymization a k-set about human physical attributes contains 10 records, 9 of which have a weight over 100kg. If an attacker knows their target is in that k-set they can be 90% certain that their target has a weight over 100kg. This means that a privacy disclosure risk is still present.

In this version of the model, we focus on *value prediction* risk. Note that although alternatives to k-anonymization, for example l-variance, may eliminate the danger of value prediction, this framework is designed to provide an assessment of any pseudonymization parameters that the user wishes to use. As stated, this framework does not attempt to produce a definitive pseudonymization technique.

The system will automatically discover and add to the LTS, transitions that correspond to a possibility of unauthorized value prediction due to pseudonymization, referred to as *risk-transitions*. A risk that a given actor (`actor.A`) can access a given sensitive field (`field.F`) is said to be present in every node in the LTS where the pseudonymized version of `F` (`F.anonymized`) has been accessed by `A`. If `A` only has access rights to `F.anon` and not `F`, *Risk-transitions* will be added to the LTS starting from each of these at-risk nodes. These will be marked as not allowed and it will be possible to calculate risk scores or declare policy associated with these transitions. Each risk transition is uniquely identified by the set of quasi-identifier fields which the actor has already accessed.

3.2 Scoring Risk Transitions - Violations

To complete the generated privacy model, a numeric value or values are calculated and added to the Risk Transitions; these state how concerned a user should be about this transition. To consider this, first a number of terms need to be defined:

- **Risk** is the danger of an individual value associated with an individual user being predicted.
- **Violations** are *risks* within the dataset which are higher than an acceptable *threshold*.
- A **Threshold** is a numeric limit above which a risk is judged to be unacceptable.
- A **Margin** specifies how close values in a continuous field are required to be to be judged to be equal for the purpose of risk calculation.

Thresholds are associated with individual values and every single value may have a different threshold. This is because they are calculated from a combination of overall system policy and an individual user's policy. Initially, an individual's *preference level* (unconcerned, pragmatist or fundamentalist as mentioned in the previous section) will dictate how high the default of their thresholds for each field should be set. Once these have been defined a user may specify particular sensitivities attached to certain fields and this will decrease the threshold value for those fields. Overall system policy will define how these user dependent factors influence thresholds. Non expert users often state preferences about information privacy that is directly opposite to their actions [13], and it cannot be assumed that they are fully aware of the implications of revealing their data. Hence, it is the responsibility of the system

designer to ensure that overall system policy does not allow thresholds outside of an acceptable range regardless of user preference.

The score used is simply the total count of violations within the data. It is calculated using the marginal probabilities of values within k-sets as illustrated in the pseudocode in Algorithm 1. Note that this process is associated with a single risk transition and therefore, as risk transitions are associated with the danger of accessing a single sensitive field, what we are calculating is the violation count for just one sensitive field (denoted v as the sensitive field corresponds to a one dimensional array of *values*).

The following terms are used in this pseudocode:

- $recordSet$. The complete set of records.
- F is the set of fields that the attacker has access to.
- k^F-set. This is the set of records which appear to be identical given the information available. In k-anonymization the k-set is the set of records that appear to be identical when the quasi identifier fields are pseudonymized. The k^F-set is the set of records which appear to be identical in the pseudonymized data when all quasi identifiers except for those in set F are masked.
- k^F-set_i is the k^F-set which record i belongs to. The size of this set is also referred to as $|k^F\text{-}set_i|$
- $|matches(k^F - set, R_i)|$ is the number of records within k^F-set_i where the sensitive value in question matches the real value in record i to within $margin$.
- $threshold_{rv}$ is the threshold associated with value v in record r.
- $margin_v$ is the margin associated with sensitive field/set of values s. This will be 0 if the field is not continuous.

Algorithm 1 Calculating Violations $calculate_violations(recordSet, v)$

```
for all r ∈ recordSet do
    violations = 0
    set = k^F − set_r
    risk = |matches(k^F − set, R_i)| ÷ |set|
    if risk > threshold_rv then
        violations = violations + 1
    end if
end for
return violations
```

4 Statistical Utility

Once violations have been detected they will normally need to be removed. It may be possible to leave some violations in the data depending on whether policy defines hard restrictions or soft restrictions. Removing will affect the statistical properties of the data and so will affect the results of experiments that researchers carry out on the data. The issue is compounded by the fact that it is harder to protect the privacy of outliers in the data [14]. Because of this, violations are not likely to be evenly spread across the data distribution which increases the possible impact on the data of removing them. To address this, when data-sets are applied to the model transitions a *utility report* for each risk transition is produced that shows both the violation count and a utility score which gives an indication of the impact that removing data has had on the dataset's statistical utility.

Table 1 Statistical values used in Utility Report

Name	Definition
maximum	The maximum value in the data-set
minimum	The minimum value in the data-set.
skewness	The asymmetry of a frequency-distribution curve.
kurtosis	The sharpness of the peak of a frequency-distribution curve.
median	The middle value in a data-set's values.
mean	The average value of the data-set.
standard deviation	The measure of spread or dispersion of a data-set.

4.1 Statistics in the Utility Report

It is impossible to define what being usable from a research point of view actually entails as this depends on the purpose for which the data is intended. The intention in our system is that the data from which violations have been removed, the *trimmed data*, is sufficiently similar to the *original data* that the difference in results from any test or statistical analysis is not sufficient to affect conclusions. This is impossible to guarantee and so our approach is to remove as few records as possible and to provide a list of statistics for both the original data and the trimmed data that is reasonably comprehensive and enables the data controller to make their own decision. The use of a standard set of statistics may also provide some reassurance. It is important to note, that differential privacy [15] is privacy-preserving method that provides such strong privacy guarantees for certain statistical operations. The approach differs from pseudonymization in that noise is added to the results of statistical queries on the data. A system designer may choose to engineer this solution, but our tool considers cases where system designer have chosen pseudonymization methods.

With these considerations in mind it was decided that the utility report should for now provide the following information:

- How many values need to be eliminated for the number violations to reach zero. Note that this number is not necessarily equal to the number of violations. This number shall be referred to as R_v for *removed violations* and the total number of violations shall be referred to as N_v for *number of violations*.
- The complete set of statistics calculated for each dataset provided by the Apache commons statistics function[1] that includes the statistics shown in Table 1.

The assumption of independence Currently the process of violation removal and utility reporting operates on a single sensitive field. The means that a *violation* is a single value at danger of prediction rather than an entire record. When a violation is removed only the relevant value is removed and the rest of the record is left untouched. This also means that the statistics above are calculated from the vector of values associated with the relevant field. Correlations between fields may be important for research purposes and so the intention is that this will be introduced into the utility report at a later stage.

4.2 Removing violations

Removing violations is not as simple as removing every value which is in violation. This strategy may, in certain circumstances, involve the removal of an unnecessary

[1] org.apache.commons.math3.stat.descriptive.DescriptiveStatistics

Table 2 Risk values for 2-anonymization data records

Set 1		Set 2	
No	Weight (kg)	No	Weight (kg)
1	70	1	70
2	77	2	80
3	78	3	74
4	75	4	74
5	79	5	74
		6	76

number of values and may also not be sufficient to remove all violations. This is because, on one hand, removing only a subset of violations may push the number of values within a certain range below the violation threshold while, on the other hand, removing all violations reduces the size of the k-set and so may create a situation where previously non violating values become violations. Further complications are created as two or more sensitive values do not have to be exactly equal to be considered identical from a violation point of view, due to the use of *margins*. Additionally the fact that different values may have different *thresholds* attached to them based on user policies makes the problem even more complex.

Table 2 provides an example of both of these situations. The sensitive field we are operating on is the *weight* field, the *threshold* is 0.75 for all values and the *margin* is 5 kilograms. The data has been divided into 2 $k^F - sets$ as shown below. Violating values are highlighted in blue. In set 1 it is only necessary to remove one value for violations to be eliminated. If all violations are removed then not only will an unnecessary number of values be removed, but the set will only have one remaining member, item 1. This will now be classed as a violation as it is, by definition, identical to 100% of members of its set.

An intuitive response might be to remove $k - n + 1$ violations at random where k is the number of violations in the set and n is the minimum number of values for a given value to be identical to (or close to given the use of *margin*) including itself in order to be considered a violation based on its threshold. For example, if the threshold is 0.75 for value v and there are 10 values in v's set then v would be in violation if it is close to 8 or more values (including itself). In this case n would be equal to 8.

There are two problems with this approach. Firstly, different values may have different thresholds based on user policy and therefore there isn't a single n value for an entire group of violations and secondly, the fact that we use margins also means that this approach would sometimes produce the wrong result.

This second problem is illustrated by set 2 in Table 2. This set contains 4 violations in a set of size 6. As we are assuming a threshold of 0.75 for all values and $4/6 < 0.75$ then k is already lower than n. According to the approach described above no values should need removing at all and the set should already be free of violations. However, the set clearly does contain violations and this is due to the use of margins. Values 3 to 5 are in violation as they are each close (within 5kg) to 5 other members of their set. They are each close to the other violating values as well as to value 1 (70kg). Value 6 is also in violation as it is close to each of the other violating values and to value 2 (80kg). Value 1 and value 2 are not in violation themselves but they must be considered in any violation removal process as their presence is responsible for other values being in violation. In this situation not just the number of violating

values removed but also which violating values are removed is important. If value 6 is removed then the set will be left with 4 violating values, all values within the range 70 to 74. If value 3, 4 or 5 is removed then the set will be left with only 2 violating values as neither a value of 70 nor 76 will lead to a violation.

As it is clearly non-trivial predicting exactly how many and which violations should be removed in all situations that may occur, the process of removing violations was made iterative. It was achieved through a modification of the $k - n + 1$ technique described above. As thresholds vary based on user preferences, a default threshold defined in the system is used to approximate how many violations should be removed. As this may not succeed in removing all violations it is followed by checking each value individually for violation and repeating this process if necessary.

Algorithm 2 describes the process in detail. The terms used in this pseudocode are defined in the following list:

- $recordSet$. The complete set of records.
- $violations[data]$. Records within the dataset $data$ containing a violation.
- F is the set of fields that the attacker has access to.
- k^F-set. This is the set of records which appear to be identical given the information available.
- K^F-set_i is the K^F-set which record i belongs to.
- $threshold$ The default threshold value that each value has as its threshold before individual user policies are applied.

As can be seen in the pseudocode, for each set violations are removed, the set is rechecked, and this is repeated until no violations remain.

Algorithm 2 Removing Violations $removeViolations(recordSet, v)$

```
do
    violations[recordSet] = calculateViolations(recordSet, v)
    for all r ∈ recordSet do
        set = K^F − set_r
        if set not already cleaned then
            numRemove_r = ((1 − threshold) ∗ |set|) − (|set| − |violations[set]|)
                ▷ randomly select (without replacement) violating values to remove and remove them
            return cleanSet(set, violations[set], numRemove_r)
        end if
    end for
while violations exist in recordSet
```

Note that, although techniques such as l-diversity [5] can eliminate the risk of value re-identification, this system is not designed to be an alternative to such techniques. It allows finer control in the situation where every single value has its own associated risk threshold. It allows greater transparency for a non expert user. We assume that a data controller has pseudonymized using a technique that they understand and have reasons for choosing. The resulting data is modified as minimally and transparently as possible in our system.

5 Pseudonymization Analysis Framework

We have implemented the prior described functionality in a software framework (developed in Java); this is available for download[2]. This section outlines the use cases for systems under design that wish to analyze the pseudonymization strategies

[2]https://github.com/operandoh2020/op-privacy-modeller

employed; and importantly the steps that a developer follows to use this tool. For this description we assume that the data-flow model has been created, and the LTS has been generated (note, this must contain at least one instance in which the data is pseudonymized for the tool to be useful). The main use cases are as follows:

1 The user, described as a *Data Controller*, is working on a live system and wishes to check the data currently in the system to ensure that an unacceptable level of policy violations isn't occurring and rethink the pseudonymization approach if it is.
2 The user, described as a *System Designer*, is designing a system and wishes to devise a pseudonymization strategy that is unlikely to lead to an unacceptable level of policy violations.
3 The user, either a *System Designer* or a *Data Controller*, wishes to impose conditions on any movement of pseudonymized data in the system so that transitions are automatically disallowed when the number of violations is above a threshold. An error will be displayed if the system attempts to send data that is a violation of these restrictions.

The overall workflow is shown in Figure 3. Actions are marked with a "UC" caption to indicate which use case they are associated with.

For use case 3, which may well be performed after use cases 1 or 2, the user provides rules of the form:

`IF` $numberOfViolations > acceptableLimit$ `[OR/AND]` $utilityThreshold < acceptableLimit$... `THEN block transition action`

There may be multiple utility threshold parameters and any combination may be used. These will generally take the form of differences between the key statistics described in Section 4.

For use cases 1 and 2 the process is essentially the same. The user starts by viewing the risk transition in the LTS of most concern to them, this is the *maximum risk transition* i.e. the transition that has the most violations. As has already been stated, risk transitions are uniquely identified by the quasi identifiers that have already been accessed and the maximum risk transition will always be the one in which all quasi identifiers have been accessed. The user will view the utility report associated with that transition and will then select an action based on that report. As an alternative to automatically viewing the maximum risk transition a user may also select risk transitions on the LTS manually.

This action may consist of either accepting the pseudonymization strategy as it is or exploring alternative pseudonymization strategies. Alternative pseudonymization strategies may involve either exploring alternative pseudonymization techniques, for example switching from k-anonymization to l-diversity, or considering allowing access to only a subset of fields. The latter option is equivalent to choosing a different risk transition. A list of all risk transitions associated with the target sensitive field will be displayed alongside the associated number of violations for each. The user selects one of these to generate the utility report associated with this new risk transition. The ability to see different subsets of fields accompanied by their violation count addresses the problem of high dimensionality in pseudonymization techniques such as k-anonymity [16]. This provides an easy to understand metric for comparing different subsets of fields and understanding which fields have the

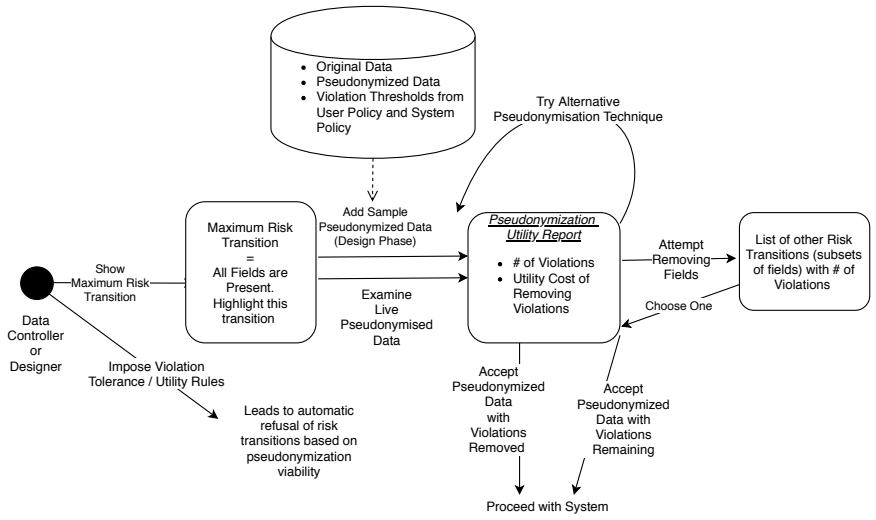

Figure 3 Overall Workflow Diagram

greatest cost in terms of re-identification risk. While most solutions attempt to address the question of whether it is possible to reduce violation cost while still sending all fields to the researcher this may not be possible and the researcher may not need all fields.

If the user chooses to accept a pseudonymization strategy they may do so either before or after violations have been removed. Clearly this depends on the nature of these violations with regard to user policy and system policy as any hard violation or violation associated with legally required policy must be removed. In a live system, accepting a pseudonymization strategy will lead to data being disclosed as the risk is judged to be acceptable. In a system being designed accepting a pseudonymization strategy will mean defining that, either before or after violation removal, this transition will be part of the system and will always occur unless this rule is subsequently changed. The only difference between use cases 1 and 2 is how the utility report is generated. In a live system it will be generated with live data while in a system being designed sample data will be used.

6 Evaluation

To evaluate we apply the framework to particular use cases and observe the extent to which a user of the framework is informed about the risks of pseudonymization. The following use cases are described in turn and show how utility and violations may be considered. For both, we prepared a health record set to undergo 2-anonymization. A researcher has access to this anonymized data but does not and should not have access to the original data. The policy violation that we wish to avoid is the researcher being able to predict an individual's weight to within 5kg (margin = 5).

6.1 Uniform Privacy Use Case

In this case we assume that the threshold is 0.9 for all users; all users are considered to have the same privacy preferences. This means that an attacker predicting any individual's weight with a 90% or above confidence is considered a violation. Age

Table 3 Risk values for 2-anonymization data records

Age	Height (cm)	Weight (kg)	Height risk	Age risk	Age Height risk
30-40	180-200	100	2/4	2/2	2/2
30-40	180-200	102	2/4	2/2	2/2
20-30	180-200	110	2/4	3/4	2/2
20-30	180-200	111	2/4	3/4	2/2
20-30	160-180	80	1/2	1/4	1/2
20-30	160-180	110	1/2	3/4	1/2
Violations:			0	2	4

and height are quasi identifiers. Table 3 provides six sample records input to the model analysis process and shows how, as more identifying fields become available to the researcher, the number of violations of this policy increases. The risk columns show the proportion of records with matching quasi identifiers (considering only those that the attacker has access to) where weight is within 5kg of the weight of the current record. If this proportion is above 0.9 then it is highlighted as a violation.

The framework generated the LTS as shown in Figure 4. Dotted lines indicate potential policy violations. A system administrator has the option of loading the six records given as examples above into the LTS. They would then see the violation scores 0, 2 and 4 as shown in this figure. A utility report would be generated showing the cost to statistics of eliminating these violations. If the number of violations or the utility cost is unacceptable the data controller can consider increasing their k value or reconsider their pseudonymization entirely. Alternatively, at the design phase, a system designer could declare that a number of violations above 50% is unacceptable. The system would now throw an error if the above data was used, forcing the administrator to choose another form of pseudonymization.

Note that this situation assumes a uniform threshold for all users. If this is not the case the number of violations will change. Suppose, for example, that we know from a user questionnaire that the owner of row 4 is a data fundamentalist while all other users are pragmatists. The system designer may have created a rule that fundamentalists should, by default, have their threshold level set to 0.7 rather than 0.9. In row 4 Risk w/height is at 3/4. This is not a violation in the default situation as it is below 0.9 but now that the user is a fundamentalist it becomes a violation. Similarly, the owner of row 3 may be a pragmatist but may have declared that weight is an especially sensitive field for them. The system designer may have created a rule that especially sensitive fields when their owner is a pragmatist also have a threshold of 0.7, leading to the same outcome for row 3.

6.2 Realistic Privacy Distribution

This use case features a larger dataset with 1103 realistic records; we show that the framework can be utilised to make decisions about data disclosure taking into account user privacy preferences.

6.2.1 Generating Sample Data

The data used was randomly generated to approximate data from the United States national health survey of 2007-2010 [17]. Among other things, this survey provides

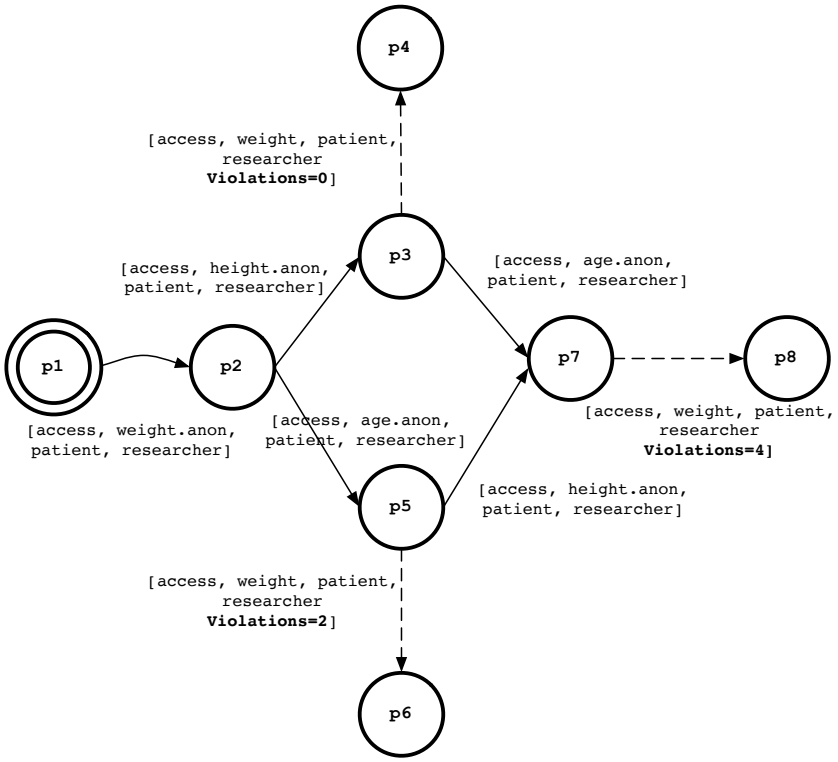

Figure 4 Pseudonymization risk analysis output

the height, BMI (Body Mass Index), sex, race (including non hispanic white, non hispanic black and hispanic) and age of 11039 adults categorised by race, sex and age. For each of these categories the mean height and BMI and the standard for each is provided. We have generated a dataset that is 10% of the size of the original survey using the same distributions. The number of individuals in each category in our dataset is exactly 10% of the number in the original data and for each category its own unique mean and standard error is used to generate these individuals. Table 4 describes our dataset. In this case weight is the sensitive field and age, sex, race and height are all quasi identifiers.

Policy Distributions Each record is associated with an imagined individual through a one to one mapping and each imagined individual has their own privacy level. Some of these hypothetical data owners also consider the weight field to be extra sensitive. The 3 by 2 matrix given in Table 5 shows how a user's privacy level controls the threshold used both for fields they consider to be extra ''sensitive" and for ''normal" sensitive fields in our example. The number of users that fall into each sensitivity category is taken directly from our own survey of the general population [3]. Table 6 shows the proportion of users falling into each category in our data and also shows how many consider weight to be an extra sensitive field.

[3]details of this user privacy study is currently under submission

Table 4 Data Distribution

sample size	sex	age range	race	mean height (cm)	height standard error	mean bmi	bmi standard error
80	male	20 -39	white	178.4	0.35	27.7	0.25
83	male	40 -59	white	178.3	0.28	29.2	0.22
110	male	60+	white	174.6	0.22	29.2	0.18
36	male	20 -39	black	176.9	0.39	28.7	0.39
37	male	40 -59	black	176.7	0.53	29.4	0.38
36	male	60 +	black	174.4	0.42	28.8	0.32
57	male	20 -39	hispanic	171.1	0.48	28.5	0.33
58	male	40 -59	hispanic	170.3	0.36	29.5	0.24
39	male	60 +	hispanic	167.3	0.45	29.2	0.32
82	female	20-39	white	164.9	0.25	27.5	0.41
86	female	40-59	white	163.8	0.27	28.3	0.24
108	female	60+	white	160.3	0.22	28.7	0.2
40	female	20-39	black	163.7	0.32	31.4	0.46
38	female	40-59	black	163.5	0.38	33.1	0.49
37	female	60+	black	160.6	0.28	31.1	0.33
67	female	20-39	hispanic	158.2	0.23	28.8	0.23
58	female	40-59	hispanic	157.1	0.33	30.2	0.34
51	female	60+	hispanic	153.7	0.31	29.9	0.17
total=1103							

Table 5 Overall Policy

	unconcerned	pragmatist	fundamentalist
normal	1	0.9	0.8
sensitive	0.9	0.8	0.7

6.2.2 Results

The sample dataset was pseudonymized initially using k-anonymity (carried out using the arx software [18] and using its default settings). As expected, a utility report was first generated for the transition in which every quasi identifier had been accessed. A screenshot of this report in the k-anonymity scenario is shown in Figure 5.

In this case 116 violations were detected and 117 values (about 1% of values) needed to be removed. This has had no effect on the maximum and minimums and has had a slight effect on the standard deviation, the skew and kurtosis. The data controller or system designer may decide that this is an acceptable statistical impact and improve the sending of the data with violations removed, they may approve the sending of the original data or they may opt instead to try an alternative pseudonymization algorithm.

Removing Fields Choosing the "try removing fields" option will result in a dialog box as shown in Figure 6. This dialog box allows the data controller to generate a utility report for a smaller subset of fields and clearly shows how variable the number of violations are for different subsets. Even for subsets containing three out of the four fields this varies from 29, if only age is unknown, to 107, if sex is unknown. In this case it is perhaps not surprising that sex has little impact on the effectiveness of pseudonymization as it only contains two categories while age contains a large

Table 6 User Policy Distribution

Pragmatist	49%
Fundamentalist	30%
Unconcerned	21%
Regard weight as sensitive	10%
Regard weight as normal	90%

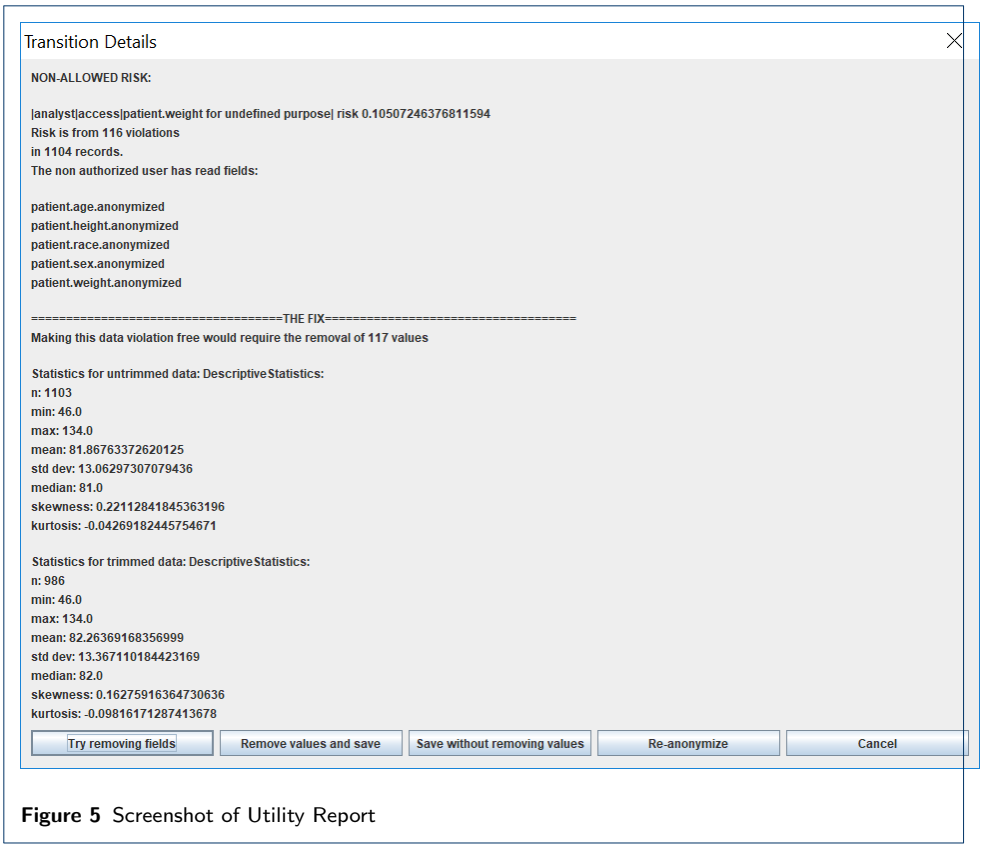

Figure 5 Screenshot of Utility Report

number of unique values and so would be expected to contribute significantly to disclosures risks. It may be more significant in this case that removing height leaves 38 violations and so has somewhat less of an impact than age, possibly due to a greater correlation between height and weight. It is perhaps most interesting that removing race has a much greater impact than removing sex (down to 67 violations) despite only containing 3 categories and despite sex being, one would assume, correlated to weight to a greater extent than race. This may be due to the unequal numbers of individuals in each racial group.

The usefulness of this information will of course depend on the purpose of the data and whether any fields can reasonably be removed. It could, however, also be used to inform decisions such as weight placed on different fields when tuning pseudonymization algorithms or indicate which fields may need their hierarchical categories rethought. These results may seem relatively straightforward and predictable in this example but this functionality becomes more useful as the number of fields increase and in situations where removing, partially obscuring or reconsidering multiple fields at once is a possibility. This is especially true with multiple fields as they may interact with each other in unpredictable ways.

6.3 Comparing with l-diversity and other algorithms

In this use case we assess the extent to which the framework can be used to make decisions about the pseudonymization algorithm chosen. An alternative pseudonymization algorithm that a data controller may consider is l-diversity. Unlike k-anonymity, l-diversity [5] protects against value prediction by guaranteeing a

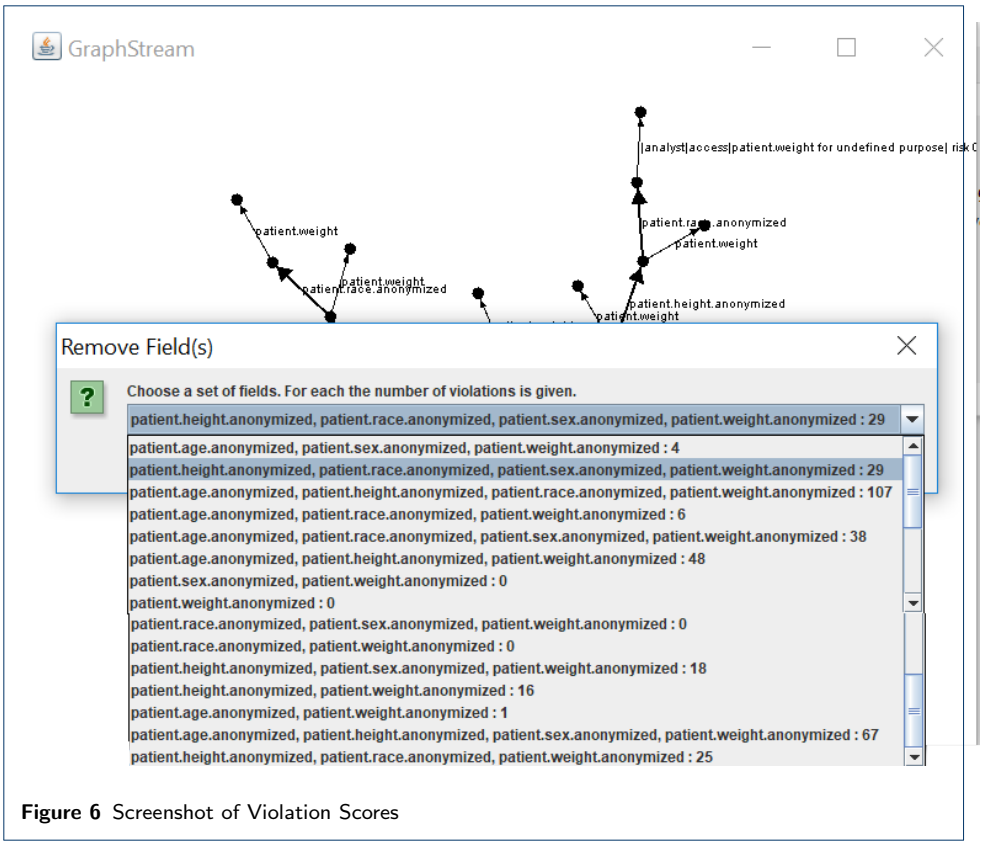

Figure 6 Screenshot of Violation Scores

Table 7 Risk values for 2-anonymization data records

Statistic		k-anonymity	l-diversity
Number of Violations		116	67
Number Removed		117	68
	Original	α	
Minimum	46	0	0
Maximum	134	0	0
Mean	81.87	-0.39	-0.32
Standard Deviation	13.06	-0.31	-0.15
Median	81	1	1
Skewness	0.22	0.06	0.05
Kurtosis	-0.04	0.06	0.03

range of different values in every set. Entropy based l-diversity goes one step further in ensuring that no single value dominates so as to protect against probabilistic predictions. Using l-diversity would therefore seem a logical strategy if the framework has revealed that using k-anonymity results in a large number of violations. For this reason we applied l-diversity using Shannon entropy to the data, again using the arx framework and default settings.

As before, a utility report was generated initially on the scenario where all fields have been read. As the key metric of utility is the difference between statistics calculated from the unmodified data and statistics from data which has been pseudonymized and had violations removed, the differences (or α) between each pair of statistics was recorded. This was done for both k-anonymity and l-diversity in order to compare the utility loss between the two algorithms. The results are shown in Table 7.

As would be expected, less violations need to be removed when l-diversity is used. However, the overall effect on utility is similar between the two algorithms. l-diversity also doesn't eliminate the need to remove violations altogether and the number of violations removed is still more than half the number removed when using k-anonymity. This is despite the fact that l-diversity, Shannon entropy l-diversity in particular, is designed for precisely this scenario when we are attempting to eliminate value predictions. Note that probabilistic value predictions in particular are the focus of this system and entropy based l-diversity is not expected to eliminate probabilistic value predictions and so l-diversity based on recursion, which is designed for this purpose, was also used [5]. This did not work for our data. Using the default arx settings it failed to produce any results. t-closeness was also tried and this, similarly, did not produce any results [19]. Both of these techniques would presumably require fine tuning to be effective and how to do this is not always clear. In our scenario values having different probability thresholds complicates the issue of avoiding probabilistic value predictions. Eliminating value prediction to the probability threshold required by fundamentalist users on their most sensitive fields would be so restrictive if applied to every value that l-diversity would be unlikely to be successful. The use of a fixed margin to define equality in continuous values may further complicate matters. More generally, we are operating on a large dataset with a lot of values for height, weight and age, several quasi identifiers and a lot of correlation between fields and so it seems likely that more restrictive pseudonymization algorithms would tend to need tuning before being successful.

This study has shown our framework's ability to give detailed information on which pseudonymization strategy is preferable to just choosing an individual strategy. This demonstrates that in a situation such as this one simply choosing a more restrictive algorithm such as l-diversity or t-closeness is not sufficient for protecting user privacy.

7 Related Work

The system discussed here incorporates pseudonymization into a wider framework. As we aim to provide novelty in how an easy to use but thorough pseudonymization framework is integrated into a wider privacy framework this section will discuss both comparable pseudonymization techniques and comparable privacy frameworks from the literature.

7.1 Pseudonymization and Differential Privacy

Many pseudonymization algorithms have been developed beginning with k-anonymity [20] [4]. This method divides data into groups such that no record is uniquely identifiable. It does not, however, address the issue that groups may contain only a single value for a sensitive field and so fails to eliminate the risk of value prediction. It also suffers from the curse of high dimensionality. With too many fields dividing data into sets can become impossible [16].

To address the issue of value prediction l-diversity was proposed by Machanavajjhala et al [5]. This extends k-anonymity by also ensuring that each individual sensitive value is well represented in each set. l-diversity guarantees that each q-set will contain at least l values for a sensitive field. This is, however, insufficient to

protect against probabilistic attacks. Although multiple values are guaranteed to exist in each q-set there is no guarantee that a single value won't still dominate. To address this, alternative versions of l-diversity such as entropy l-diversity and recursive l-diversity that aim to ensure that no value dominates. This objective does, however, remain hard to achieve in practice and may be overly limiting. t-closeness ensures that the distribution of an attribute is each sensitive attribute in each set is close to its distribution in the overall table [19]. A limitation of t-closeness is that it does not protect against identity disclosure.

Alternatively, pseudonymization may be achieved through perturbation. Perturbation involves creating new values from the same distribution as the original values [21]. This suffers from the drawback that it assumes independence between variables and so potentially very useful correlation information is lost to researchers [22]. Perturbation methods also do not guarantee that a record is indistinguishable from a quantifiable number of other records in the way that k-anonymity does [22].

All of these approaches suffer from the disadvantage that they do not take account of individual data subjects having different requirements for different sensitive fields. In order to address this the concept of *personalized privacy protection* was developed by Xiao and Tao [23]. This approach is similar to the work described in this paper as it involves soliciting information from each user as to which sensitive fields are most important and giving these fields higher priority for protection. This goes some way towards what we are proposing but it defines sensitive fields in terms of their sensitivity relative to other fields. It does not allow the integration of a system of numeric sensitivity levels or allow for users to have a low or high sensitivity preference overall.

An alternative method for incorporating user preference is the condensation based approach proposed by Aggarwal et al [24]. In this approach the data is divided into groups of records which are guaranteed not to be distinguishable from each other as with k-anonymity. The difference is that the size of each group is determined by the sensitivity level of the most sensitive value within it. That is to say, a value that a data subject specifies as highly sensitive will be allocated to a large group to minimise the probability of value prediction. This solution faces potential efficiency issues as less sensitive values may be placed into groups with more sensitive values and therefore may receive a higher level of pseudonymization than necessary. This method also uses perturbation and so involves modifying sensitive values, possibly to a greater extent than is required.

In terms of the presentation of pseudonymization for users there are a number of tools available to anonymize data, which also provide some risk analysis feedback. The ARX Tool [18] provides methods for analyzing re-identification risks following the prosecutor, journalist and marketer attacker models on a number of anonymization algorithms. The Cornell Anonymization Toolkit (CAT) [25] performs Risk Analysis by evaluating the disclosure of risks of each value in pseudonymized data based on user specified assumptions about the adversary's background knowledge. These tools offer important insights to identify privacy risks; and in our approach we seek to integrate similar capabilities (alongside fine-grained user privacy consideration) into our privacy-by-design methodology for developing distributed data services.

Differential privacy [15] is a technique that provides strong formal guarantees of the privacy of users where personal data is released from statistical databases. Rather than changing the dataset itself (as with pseudonymization methods), differential privacy adds noise to the output of queries performed on the data. This guarantees that for any query, or sequence of queries, a subject and their personal data cannot be identified. Hence, it is an alternative method to achieving the same results as using the pseudonymization framework of this paper. However, there are downsides in the face of such strong privacy guarantees; i.e. the difficulty in developing the correct noise functions, the cases where the need to add too much noise reduces the statistical utility of the data, and also the situations where data is released without knowing what functions will be performed on it. Hence, developers will continue to consider pseudonymization methods, which this tool supports. There are tools and frameworks to help non-experts carry out differential privacy e.g. PSI [26] and Adaptive Fuzz [27], therefore an interesting avenue of future research is to consider privacy requirements and risk across the differing methods.

7.2 Privacy Frameworks

Both Fischer [11] and Kosa [10] define formal models of privacy in terms of state machine representations. Their purpose to demonstrate that a system complies with privacy regulations and requirements. Such models offer strong building blocks that our formal privacy model builds upon; in particular moving from hand-crafted specifications to auto-generated models that underpin the privacy engineering process and privacy risk analysis. MAPaS [28], is a model-based framework for the specification and analysis of privacy-aware systems. Centred upon a UML Profile, purpose-based access control systems are modelled and the framework allows queries to be executed to identify errors in the design.

LINDDUN [29] is a framework for performing privacy threat analysis on the design of a system in order to select privacy enforcing mechanisms that mitigate the threats. This combines a data flow model of the system with a privacy threat catalogue to provide a design-time methodology to assess privacy risks. We similarly employ a data-flow oriented methodology but explore the extent risk can be analysed automatically via the generation of an underpinning formal model. Further, we consider the use of MDE methods beyond the design phase (and in particular analysis of running systems with real users).

A system's behaviour should be matched against it's own privacy policy. [30] models a system's behaviour in terms of a Business Processing Model Notation (BPMN) diagram and then the goal is to check whether this is compliant with the system's P3P privacy policy. [31] integrate links to the privacy policy in the system's workflow (e.g. the BPEL specification), these are then checked by an analysis tool at design time to determine if the workflow agrees with the policy. [6] provide a similar method; rather than having a designer merge the workflow and policy, the approach converts both models (a BPEL specification and P3P policy) into a graph representation before formally analyzing the correctness of the graph. However, all of these solutions only check if a system behaves according to its stated privacy policy (our LTS can be similarly analysed); there has been limited research into the evaluation of a system in terms of privacy risk considering fine-grained user preferences.

8 Conclusion

This paper has discussed the pseudonymization aspect of a software framework for measuring privacy risk in a multi actor system. It identifies pseudonymization risk and provides easily understandable information to help a user without expert knowledge choose a pseudonymization strategy. The concept of individual policy is central to this system to ensure that the preferences of individual data subjects are taken into account. Risk is quantified based on policy violation and also based on what impact removing these violations will make from a statistical point of view. Transparency is also key. By integrating these concerns into pseudonymization and integrating pseudonymization into a larger privacy framework the system developed goes beyond existing pseudonymization techniques. We have demonstrated through anexperimental evaluation) that, in variable policy scenarios, simply choosing a more thorough pseudonymizing techniques is not sufficient to eliminate violations and so more information is needed to help the user choose a pseudonymization strategy. This system may be useful either for a system designer designing a system or for a data controller supervising a system.

Future Work. A set of standard pseudonymization techniques could be provided so that the user doesn't have to manually input pseudonymized data although with the existence of more complete tools such as arx this may not be necessary. Challenges here concern the huge number of tunable parameters involved in the pseudonymization process. Utility metrics will also ultimately be improved. In particular correlations will be incorporated as correlations between fields is often more important in data analysis than the statistical properties of individual fields. Risk (and consequently violation) calculations can be improved by incorporating more situations, such as when an attacker does not know the value of a quasi identifier exactly for their target but access to this quasi identifier, pseudonymized or not, may still provide clues for value prediction.

We are also investigating the integration of the rule-based decision system (predicated on the utility reports) into distributed systems software; such as privacy-oriented middleware and cloud protection systems. Enabling fine-grained user privacy enforcement where the system automatically makes decisions as to whether a disclosure of pseudonymized data will be allowed for any occurring distributed system action.

Author details

[1]IT Innovation, University of Southampton, Gamma House, Enterprise Road, SO16 7NS Southampton, UK. [2]IT Innovation, University of Southampton, Gamma House, Enterprise Road, SO16 7NS Southampton, UK. [3]IT Innovation, University of Southampton, Gamma House, Enterprise Road, SO16 7NS Southampton, UK. [4]IT Innovation, University of Southampton, Gamma House, Enterprise Road, SO16 7NS Southampton, UK.

References

1. Kumaraguru, P., Cranor, L.F.: Privacy indexes: a survey of westin's studies (2005)
2. Acquisti, A., Brandimarte, L., Loewenstein, G.: Privacy and human behavior in the age of information. Science **347**(6221), 509–514 (2015)
3. Hildebrandt, M.: Profile transparency by design? re-enabling double contingency, (2013)
4. Sweeney, L.: k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **10**(05), 557–570 (2002)
5. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: L-diversity: privacy beyond k-anonymity. In: 22nd International Conference on Data Engineering (ICDE'06), pp. 24–24 (2006)
6. Li, Y.H., Paik, H.-Y., Benatallah, B.: Formal consistency verification between bpel process and privacy policy. In: Proceedings of the 2006 International Conference on Privacy, Security and Trust, p. 26 (2006). ACM
7. Hadar, I., Hasson, T., Ayalon, O., Toch, E., Birnhack, M., Sherman, S., Balissa, A.: Privacy by designers: software developers' privacy mindset. Empirical Software Engineering **23**(1), 259–289 (2018)

8. Cavoukian, A.: Privacy by design. Take the challenge. Information and privacy commissioner of Ontario, Canada (2009)
9. Grace, P., Burns, D., Neumann, G., Pickering, B., Melas, P., Surridge, M.: Identifying privacy risks in distributed data services: A model-driven approach. In: Proceedings of the 2018 IEEE International Conference on Distributed Computing Systems. ICDCS '18 (2018)
10. Kosa, T.A.: Towards measuring privacy. PhD thesis, University of Ontario Institute of Technology (April 2015)
11. Fischer-Hübner, S., Ott, A.: From a formal privacy model to its implementation. In: Proceedings of the 21st National Information Systems (1998)
12. Grace, P., Surridge, M.: Towards a model of user-centered privacy preservation. In: Proceedings of the 12th International Conference on Availability, Reliability and Security. ARES '17, pp. 91–1918. ACM, New York, NY, USA (2017)
13. Norberg, P.A., Horne, D.R., Horne, D.A.: The privacy paradox: Personal information disclosure intentions versus behaviors. Journal of Consumer Affairs **41**(1), 100–126
14. Hay, M., Miklau, G., Jensen, D., Towsley, D., Weis, P.: Resisting structural re-identification in anonymized social networks. Proceedings of the VLDB Endowment **1**(1), 102–114 (2008)
15. Dwork, C.: Differential privacy: A survey of results. In: Agrawal, M., Du, D., Duan, Z., Li, A. (eds.) Theory and Applications of Models of Computation, pp. 1–19. Springer, Berlin, Heidelberg (2008)
16. Aggarwal, C.C.: On k-anonymity and the curse of dimensionality. In: Proceedings of the 31st International Conference on Very Large Data Bases, pp. 901–909 (2005). VLDB Endowment
17. Fryar, C.D., Gu, Q., Ogden, C.L.: Anthropometric reference data for children and adults: United states, 2007-2010. Vital and health statistics. Series 11, Data from the national health survey (252), 1–48 (2012)
18. Prasser, F., Kohlmayer, F.: In: Gkoulalas-Divanis, A., Loukides, G. (eds.) Putting Statistical Disclosure Control into Practice: The ARX Data Anonymization Tool, pp. 111–148. Springer, Cham (2015)
19. Li, N., Li, T., Venkatasubramanian, S.: t-closeness: Privacy beyond k-anonymity and l-diversity. In: 2007 IEEE 23rd International Conference on Data Engineering, pp. 106–115 (2007)
20. Samarati, P., Sweeney, L.: Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, Technical report, SRI International (1998)
21. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: ACM Sigmod Record, vol. 29, pp. 439–450 (2000). ACM
22. Aggarwal, C.C., Philip, S.Y.: A condensation approach to privacy preserving data mining. In: International Conference on Extending Database Technology, pp. 183–199 (2004). Springer
23. Xiao, X., Tao, Y.: Personalized privacy preservation. In: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data. SIGMOD '06, pp. 229–240. ACM, New York, NY, USA (2006)
24. Aggarwal, C.C., Yu, P.S.: On variable constraints in privacy preserving data mining. In: Proceedings of the 2005 SIAM International Conference on Data Mining, pp. 115–125 (2005). SIAM
25. Xiao, X., Wang, G., Gehrke, J.: Interactive anonymization of sensitive data. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data. SIGMOD '09, pp. 1051–1054. ACM, New York, NY, USA (2009)
26. Murtagh, J., Taylor, K., Kellaris, G., Vadhan, S.: Usable Differential Privacy: A Case Study with PSI. ArXiv e-prints (2018).
27. Winograd-Cort, D., Haeberlen, A., Roth, A., Pierce, B.C.: A framework for adaptive differential privacy. Proc. ACM Program. Lang. **1**(ICFP), 10–11029 (2017)
28. Colombo, P., Ferrari, E.: Towards a modeling and analysis framework for privacy-aware systems. In: 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing, pp. 81–90 (2012)
29. Deng, M., Wuyts, K., Scandariato, R., Preneel, B., Joosen, W.: A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. Requirements Engineering **16**(1), 3–32 (2011)
30. Chinosi, M., Trombetta, A., *et al.*: Integrating privacy policies into business processes. Journal of Research and Practice in Information Technology **41**(2), 155 (2009)
31. Short, S., Kaluvuri, S.P.: A data-centric approach for privacy-aware business process enablement. In: International IFIP Working Conference on Enterprise Interoperability, pp. 191–203 (2011). Springer

Declarations

List of abbreviations
MDE: Model Drive Engineering
LTS: Labelled Transition System
BMI: Body Mass Index
CAT: Cornell Anonymization Toolkit
BPMN: Business Processing Model Notation.
BPEL: Business Process Execution Language

Ethics approval and consent to participate
Not applicable

Consent for publication
Not applicable

Availability of data and materials
The software and data supporting the conclusions of this article are available from https://github.com/OPERANDOH2020/op-privacy-modelling

Competing interests
The authors declare that they have no competing interests.

Author's contributions
GN conceived the pseudonymization risk analysis algorithms, and designed and performed the experiments. PG, DB, MS and GN conceived and developed the underlying user privacy model theory and framework design. GN and PG implemented the software framework used in the paper. MS supervised the research project. All authors discussed the results, contributed to, and approved the final manuscript.

Acknowledgements
This work was supported by the European Commission under the Horizon 2020 Programme (H2020), as part of the OPERANDO project (Grant Agreements no. 653704).