Anonymity: Formalisation of Privacy – k-anonymity

Janosch Maier
Betreuer: Ralph Holz
Seminar Future Internet SS2013
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: maierj@in.tum.de

ABSTRACT

Microdata is the basis of statistical studies. If microdata is released, it can leak sensitive information about the participants, even if identifiers like name or social security number are removed. A proper anonymization for statistical microdata is essential. K-anonymity has been intensively discussed as a measure for anonymity in statistical data. Quasi identifiers are attributes that might be used to identify single participating entities in a study. Linking different tables can leak sensitive information. Therefore k-anonymity requires that each combination of values for the quasi identifiers appears at least k times in the data. When subsequent data is released certain limitations have to be followed for the complete data to adhere to k-anonymity. In this paper, we depict the anonymity level of k-anonymity. We show, how l-diversity and t-closeness provide a stronger level of anonymity as k-anonymity. As microdata has to be anonymized, free toolboxes are available in the internet to provide k-anonymity, l-diversity and t-closeness. We present the Cornell Anonymization Toolkit and the UTD Anonymization Toolbox. Together with Kern, we analyzed geodata gathered from android devices due to its anonymity level. Therefore, we transferred the data into an sqlite database for easier data manipulation. We used SQL-queries to show how this data is not anonymous. We provide a value generalization hierarchy based on the attributes model, device, version and network. Using the UTD Anonymization Toolbox, we transferred the data into a k-anonymous state. For different values of k there are different possibilities of generalizations. We show parts of a 3-anonymous version of the input data in this paper.

Keywords

android, anonymity, k-anonymity, re-identification, privacy

1. INTRODUCTION

Companies gather data to provide their customers with tailored advertisements. Public institutions collect data for research purposes. There is census data medical data and data on economical evolution. Large amounts of gathered data are available for researchers, third companies or the public. Gathered data can be divided into microdata and macrodata. Microdata is the data in its raw form where each dataset represents one participant. In a census study this could be a person or household. Macrodata refers to aggregated data such as statistical analysis of the microdata.

The German data privacy act states that data is anonymized if identification of single people or entities is difficult or impossible [1]. If microdata is not properly anonymized, it is possible to identify single people out of a large dataset. This can be achieved by linking a table against data in another table or simply using knowledge about a single person. Using the attributes age, birth date and zip-code, Massachusetts medical data was matched against the voters registration list. As Sweeney has shown in her paper, this lead to the identification of Massachusetts Governor's medical data [12]. She proposes k-anonymity which protects against such attacks.

Example 1. *Fictitious microdata of a census study.*

	Quasi-Identifier			Sensitive Data
	Age	Gender	ZIP	Income
1	35	Male	81243	300,000
2	48	Female	83123	30,000
3	40	Male	81205	1,000,000
4	60	Male	73193	100,000
5	27	Female	83123	60,000
6	60	Male	71234	20,000
7	27	Female	83981	25,000
8	35	Female	83012	30,000
9	27	Male	81021	40,000
10	46	Male	73013	25,000
11	46	Female	83561	70,000
12	40	Male	81912	40,000
13	48	Male	72231	1,500,000

Table 1: Private table P

The private table P as shown in table 1, holds no single field that identifies a participant of the study. The income is regarded as sensitive data. It should not be possible to identify the income of a single person, using this table. If certain data such as age, gender and zip-code are unique they might be used to identify one person and his income. They can be used to match the data against a table containing age, gender and zip-code as well as the name. If the attacker is interested in a person of which he already knows the particular fields, this is even easier.

To prevent unauthorized access to information in databases, Denning and Lunt [2] describe multilevel databases in conference proceedings. Multilevel databases provide access control on different views of data. This access control is based on the security classification of the data and the security clearance of the accessing entity. If more than one data

doi: 10.2313/NET-2013-08-1_06

holder is involved and data is classified differently across the data holders, the overall classification cannot be guaranteed. Linkage of the partially available data might be sufficient to recreate the original data.

The following chapter introduces k-anonymity and how it protects data against linking-attacks. Chapter 3 presents two anonymization toolboxes. In chapter 4 data gathered using android applications is presented. Due to its structure this data provides no anonymity. This structure is assessed and described. A way to provide k-anonymity for this data is shown. Chapter 5 puts this paper in contrast to similar work. The last chapter concludes this paper and embraces the findings of this paper.

2. DESCRIPTION OF K-ANONYMITY

Data anonymization is a topic with several current studies. In [12] an approach called k-anonymity is proposed. The following sections describe the terms and notations used and introduces k-anonymity.

2.1 Working with databases

This chapter gives implications of using relational databases as basis for anonymity evaluation.

2.1.1 Relational databases

In this paper data means personal information that is organized in a table-like scheme. Each row is called tuple and contains a set of information associated with one person. Columns partition the data into semantic categories called attributes. A dataset refers to a single tuple in a particular table. The textbooks of Kemper [4] or Ullman [13] provide an elaborate description of relational databases.

To be compliant with the notation in [12] a table T is noted as $T(A_1, ..., A_n)$ with its attributes $\{A_1, ..., A_n\}$. An ordered n-tuple $[d_1, d_2, ..., d_n]$ contains the values associated with the tables' attributes. For each $j = 1, 2, ..., n$ the value of d_j is assigned to the attribute A_j.

$T[A_i, ..., A_j]$ means the projection of T, only including the attributes $A_i, ..., A_j$. Duplicate tuples are kept within the projection.

2.1.2 Quasi identifiers

A set of attributes "that are not structural uniques but might be empirically unique and therefore in principle uniquely identify a population unit" [10] is called a quasi identifier in a glossary issued by several statistical institutes.

U is a population whose data is stored in a table $T(A_1, ..., A_n)$ and a subset of a larger population U'. $f_c : U \to T$ is a function that maps the population to the table and $f_g : T \to U'$ a function mapping information from the table back to a base population. f_c can be a questionnaire in a study asking for certain attributes of the participants. f_g can be a checkup in a telephone book using certain attributes to identify the owner of a dataset.

A quasi identifier of T is defined as follows: Q_T is a quasi identifier if $\exists p_i \in U | f_g(f_c(p_i)[Q_T]) = p_i$. Verbally, a set of attributes is a quasi identifier if it is sufficient input for the checkup function f_c to uniquely identify at least one participant as the owner of a particular tuple.

In [12], Sweeney assumes that the data holder can identify attributes that might be available in external information. Therefore he can identify attributes within his data as quasi-identifiers.

EXAMPLE 2. *Quasi identifier*

A quasi-identifier for the table P from example 1 can be $Q_P = \{age, gender, zip\}$.

2.2 The k-anonymity model

A table $T(A_1, ..., A_n)$ with quasi identifier Q_T is called k-anonymous, if every combination of values in $T[Q_T]$ appears at least k times in $T[Q_T]$ [12].

EXAMPLE 3. *Table adhering to k-anonymity with $k = 4$*

	Quasi-Identifier			Sensitive Data
	Age	Gender	ZIP	Income
1	<45	Male	81***	40,000
2	<45	Male	81***	40,000
3	<45	Male	81***	300,000
4	<45	Male	81***	1,000,000
5	≥45	Male	7****	20,000
6	≥45	Male	7****	25,000
7	≥45	Male	7****	100,000
8	≥45	Male	7****	1,500,000
9	*	Female	83***	25,000
10	*	Female	83***	30,000
11	*	Female	83***	30,000
12	*	Female	83***	60,000
13	*	Female	83***	70,000

Table 2: Generalized table $G1$ based on P

To achieve k-anonymity for P, the attributes in the quasi-identifier have to be generalized. A * in the zip-code can mean any digit. In the age column <45 denotes that the age is below 45, ≥ 45 means that the age is above or equal 45 and a * as age can mean any number. $G1$ as shown in table 2 is a generalized version of P which satisfies k-anonymity with $k = 4$. Each block with the same quasi-identifier consists of at least 4 entries. If an attacker is interested in the income of a person with a certain quasi-identifier, there are at least $k - 1 = 3$ further people with the same quasi-identifier in the table.

If there are at least k tuples with the same quasi-identifier, it is not possible to identify a single tuple based on it. There are at $k - 1$ tuples with the same quasi-identifier, not distinguishable from the tuple an attacker is looking for.

2.3 Attacks against k-anonymity

Releasing several datasets based on the same group of data holders creates additional attack vectors. Three such attacks are depicted below. When releasing subsequent datasets, some accompanying practices can prevent these attacks [12].

2.3.1 Unsorted matching attack

doi: 10.2313/NET-2013-08-1_06

Two tables released can be used to link datasets, if they are based on the same original table and the position of the tuples is the same in each table. As the model of k-anonymity makes use of the relational model, theoretically there is no predefined order of the tuples. Relations are a set of tuples [13] and in sets there is no order. When real dataset are released, there is a high chance for the tuples to be ordered by some attribute. A general way is to order data ascending or descending by one or several attributes that are significant for the study. Those might be the sensitive parts of or all sensitive attributes or attributes in the quasi-identifier.

Example 4. *Unsorted matching attack*

	Quasi-Identifier			Sensitive Data
	Age	Gender	ZIP	Income
1	27	*	*****	40,000
2	40	*	*****	40,000
3	35	*	*****	300,000
4	40	*	*****	1,000,000
5	60	*	*****	20,000
6	46	*	*****	25,000
7	60	*	*****	100,000
8	48	*	*****	1,500,000
9	27	*	*****	25,000
10	34	*	*****	30,000
11	48	*	*****	30,000
12	27	*	*****	60,000
13	46	*	*****	70,000

Table 3: Generalized table $G2$

$G2$ in table 3 is based on P. The order is the same as in $G1$. $G1$ satisfies k-anonymity with $k = 4$, $G2$ with $k = 2$. If those tables are both released, the attacker can link the tuples based on their position. He is able to gain knowledge about age and gender of each tuple. This might be enough knowledge to identify single people.

This attack can trivially be prevented by randomizing the order of the tuples, when releasing datasets [12].

2.3.2 Complementary release attack

If a table is released that contains a subset of a previously released table, a complementary release attack might be possible. Attributes that are not part of the quasi-identifier of the first table can be used to link those tables.

Example 5. *Complementary Release attack*

G3 as in table 4 is an anonymized form of P. If it is released after $G1$ some tuples can be linked, even though their positions are randomized. There is only one person with an income of 1,500,000. This is sufficient to match the corresponding tuples and gain information on Q_P. The linked value for this tuple is [48, Male, 7****, 1,500,000]. For all tuples with a unique combination of values for $\{Q_{G1} \cup Income\}$ this is possible.

To prevent this attack, all attributes of the first released attack should be treated as quasi-identifier for the second table [12]. In this case this table cannot be released like this, as it does not satisfy any k-anonymity. If the second table used the previously released $G1$ as base, there would be no attack vector either. If the anonymization scheme is the same, there is no additional information that can be retrieved.

	Quasi-Identifier			Sensitive Data
	Age	Gender	ZIP	Income
1	46	*	*****	25,000
2	27	*	*****	40,000
3	48	*	*****	30,000
4	40	*	*****	40,000
5	40	*	*****	1,000,000
6	27	*	*****	60,000
7	46	*	*****	70,000
8	60	*	*****	20,000
9	60	*	*****	100,000
10	27	*	*****	25,000
11	34	*	*****	30,000
12	35	*	*****	300,000
13	48	*	*****	1,500,000

Table 4: Generalized table $G3$

2.3.3 Temporal attack

As data gathering is mostly done regularly, datasets are expected to grow over time. Changes in tuples or removal is also possible. Therefore subsequent datasets are often released. These releases can be vulnerable against linking with preceding tables [12].

Example 6. *Temporal attack*

Assume that a study has gathered data as P, and released $G1$. Later additional tuples are collected and the updated private table P_{t1} becomes: $P \cup \{$[51, Female, 83581, 28,000], [51, Male, 81019, 32,000]$\}$. Based on P_{t1} a generalized table G_{t1} as $G3 \cup \{$[51, *, *****, 28,000], [51, *, *****, 32,000]$\}$ is released. As shown in example 5, the tables $G1$ and $G3$ can be linked. Similarly the income can be used to link $G1$ and G_{t1}.

To prevent temporal attacks like this, the base for the subsequent release should be $G1 \cup (P_{t1} - P)$. In the case of example 6 the released table could be $G1 \cup \{$[*, Female, 83***, 28,000], [≥45, Male, 81***, 32,000]$\}$

2.4 Summary of k-anonymity

A table T that satisfies k-anonymity with regard to the quasi-identifier Q_T protects against re-identification of single data holders [12]. Previously released tables should be used as base for further releases. This ensures that no data leakage is possible by linking those tables. Other attacks – for example based on the distribution of sensitive attributes – may not be stopped with k-anonymity.

2.5 Further anonymity concepts

Further concepts are needed to achieve stronger anonymity.

L-diversity was described in a paper by Machanavajjhala and Kifer [8]. It takes into account that in a k-anonymous

doi: 10.2313/NET-2013-08-1_06

database all tuples in a generalized set can have the same value for the sensitive attribute. In this case the value of the sensitive attribute can be linked to each person in the set. A person in this set is not anonymous, even though there are $k-1$ other people with the same values for the quasi identifier. L-diversity has a further requirement for these tuples in such a set. The set must contain at least l different values for the sensitive attribute. This provides protection against the sketched attack.

T-closeness as presented on a conference by Li and Li [7] is a anonymity concept stronger than l-diversity. It generalizes each set in a way that the distribution of sensitive attributes of different sets differs as minimally as possible. Therefore no information can be obtained by examining the sensitive attributes in different sets. In an l-diverse table each set can contain similar but not equal attributes for a sensitive value. If an attacker can identify a person to belong to such a set, he gains knowledge of the range of the sensitive attribute, although the precise value stays unknown. T-closeness provides protection against this kind of attack.

3. ANONYMIZING TOOLBOXES

For anonymization of data there are several toolboxes and implementations of algorithms freely available. They are based on several algorithms described in scientific papers.

3.1 Cornell Anonymization Toolkit

The Cornell Anonymization Toolkit (CAT) is a Windows tool with graphical user interface. It can be used for data generalization, risk analysis, utility evaluation, sensitive record manipulation and visualization. A complete description can be found in its manual [16]. All features are applied against the data in main memory. The CAT can be used to achieve l-diversity and t-closeness. As stated on a conference by Xiao et al. [17], it uses the Incognito algorithm as described in the conference proceedings by LeFevre et al. [6] for anonymization. CAT uses several text files as input and cannot directly work upon a database.

3.2 UTD Anonymization Toolbox

The UTD Anonymization Toolbox as described in its manual [14] is a cross-platform tool running on Linux and Windows. The toolbox uses an integrated sqlite database to mitigate memory issues. The UTD Anonymization Toolbox can be used to achieve k-anonymity, l-diversity and t-closeness. It uses the Datafly algorithm proposed by Sweeney in a journal publication [11] and Incognito [6] algorithm for anonymization. The toolkit uses text files as input but other data formats as well as a database connector are planned for future releases [14].

4. EVALUATION OF ANDROID GEODATA

In Wagner's Bachelor Thesis [15] cellular networks are assessed based on data provided from android users via an android application. In this chapter the anonymity of Wagner's microdata is evaluated.

4.1 Data collection

The collected data consists of a timestamp, version and data about speed, ping, cell, gateways, wifi, location, device, network interfaces, traceroute and global IP. Attributes identifying a single user, such as G-mail address are not collected. Each dataset is identified by a 32 character hex string. Each device is assigned to a random 32 character hex string as well. A detailed description of the attributes can be found in [15]. The data is organized in json files with each file belonging to one device. A file can consist of different datasets. The complete data contains 166,960 datasets in 989 device files.

For this paper, we treat the attributes *longitude* and *latitude* as sensitive data. For better analysis these as well as the attributes *deviceId, dataId, device, model, version* and *network* were transferred to a sqlite database. The tables created are called 'android_data' and 'android_data_full'. For the first evaluation the first dataset from each device file was stored in a simplified table 'android_data' = S(*deviceId, device, model, version, network*). For later analysis all datasets are regarded and stored in the full table 'android_data_full' = F(*deviceId, dataId, device, model, version, network*). The simplified table takes into account, that an attacker might be able to establish a connection between datasets of the same device. This problem can be described, as if there was only one dataset per device. If the simplified table shows a certain level of anonymity, the anonymity of the full table is at least as good.

4.2 Anonymity Level

The tables S and F are inspected separately. $Q_S = Q_F =$ (device, model, version, device) is assumed to be a quasi-identifier for the data in S respectively F.

4.2.1 Simplified Database

Testing whether a combination of attributes is a possible quasi-identifier can be achieved by checking its uniqueness. In a relational database a SQL-command like in listing 1 can be used. This statement groups all tuples in the database which have the same values for all attributes in Q_S. All tuples where this combination of values is unique are printed. An excerpt of the result is shown in listing 2. If a device appears in that list, there is no other device with this quasi-identifier in the data.

```
SELECT device, model, version, network
  FROM android_data
  GROUP BY device, model, version, network
  HAVING count(*) = 1;
```

Listing 1: Checking potential quasi identifier

The most common values in Q_S are ('glacier', 'HTC Glacier', 8, 'T - Mobile') with 119 occurrences in the database. This means that 119 different devices in the study were HTC Glaciers with SDK version 8 and T-Mobile as network provider. Any participant with a device like this is already well protected against linking. 214 devices have a unique combination of values for Q_S. This means that each of those 214 users is the only user with this particular used device.

```
(device, model, version, network)
('cdma_solana', 'DROID3', 10, '')
('cdma_targa', 'DROID BIONIC', 10, '')
('chacha', 'HTC Status', 10, 'AT&T')
('crespo', 'Nexus S', 9, 'T-Mobile')
('crespo', 'Nexus S', 10, '')
```


doi: 10.2313/NET-2013-08-1_06

```
('crespo', 'Nexus S', 10, 'Airtel')
('crespo', 'Nexus S', 10, 'COSMOTE')
```

Listing 2: Excerpt of unique tuples in S

Table 5 shows how many tuples in S with the following condition exist: For a single combination of attributes there is only one tuple in the table. This data is visualized in figure 1. The red dots represent single values of a given quasi identifier combination with a certain number of attributes. The green dots are the arithmetic mean of the according single values.

Attributes	Unique tuples
version	0
device	52
model	52
device, model	58
network	66
device, version	66
model, version	70
device, model, version	75
version, network	93
device, network	188
model, network	191
device, model, network	195
model, version, network	210
device, model, version, network	214

Table 5: Number of unique tuples for an assumed quasi identifier

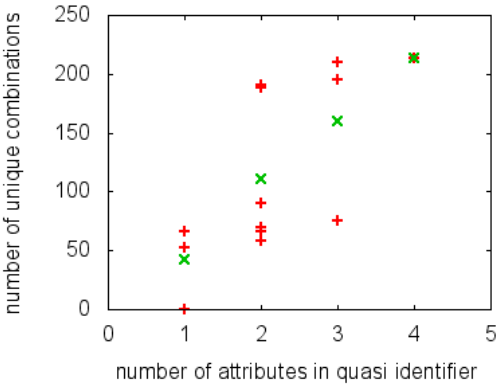

Figure 1: Visualization of unique tuples for an assumed quasi identifier

Within nearly 1,000 entries the device name is enough to identify 52 users. If device, model, version and network are taken as quasi-identifiers, over 20% of all users are uniquely identifiable.

It is unlikely that there are tables publicly available to link the possibly obtained location data based on this quasi-identifier. Nevertheless it is common to get hold of a friend's phone. Possessing an android device, it is easy to get all the needed information. Employers with a bring-your-own-device policy might track the necessary device information or provide their employees company phones. This might be exploited by attackers.

The original data is organized on a one file per device basis. When releasing the data, a similar approach might be chosen. An attacker is then able to identify which tuples belong to the same device. In this case, the anonymity level described in this chapter applies.

4.2.2 *Full Database*

The table F can contain more than one tuple for a single value of Q_F, even if it was identified as solely in S using the command from listing 1. This means that a device appears several times in the data. Different locations in those datasets are likely. Based on Q_F, an attacker cannot distinguish if two tuples with the same values for Q_F belong to the same device. For users that appear more often in the data this increases their anonymity level. Similarly for all users with the same values for Q_F the level of anonymity increases.

Similar to listing 1, the SQL-statement in listing 3 identifies which tuples in F exist that fulfill the following condition: There is no other tuple in F with the same values in Q_F. The statement returns 30 tuples. An excerpt is shown in listing 4.

```
SELECT device, model, version, network
  FROM android_data_full
  GROUP BY device, model, version, network
  HAVING count(*) = 1;
```

Listing 3: Checking potential quasi identifier in full tables

```
(device, model, version, network)
('a1', 'Acer Liquid', 8, 'ROGERS')
('ace', 'Desire HD', 10, 'SONERA')
('ace', 'Desire HD', 10, 'Saunalahti')
('bravo', 'HTC Desire', 8, 'AT&T')
('buzz', 'HTC Wildfire', 10, 'vodafone HU')
('cdma_targa', 'DROID BIONIC', 10, '')
('crespo', 'Nexus S', 10, 'COSMOTE')
```

Listing 4: Excerpt of unique tuples {device, model, version, network}

Releasing the full data without possibility to link tuples belonging to one device, an attacker can gain less knowledge, than described in chapter 4.2.1. Nevertheless he is able to retrieve accurate position data for 30 users based on Q_F.

EXAMPLE 7. *Identification based on android device features*

Assume the two friends Alice and Eve both have installed this particular android app and talked about it. Therefore Eve knows that Alice takes place in this study. She also knows that Alice has an HTC Desire with AT&T as mobile provider. Eve is able to identify Alice's tuple within the dataset and use it to create a location profile of Alice.

4.3 Anonymization

For anonymizing data, generalization and suppression are concepts used in several algorithms. This chapter presents

doi: 10.2313/NET-2013-08-1_06

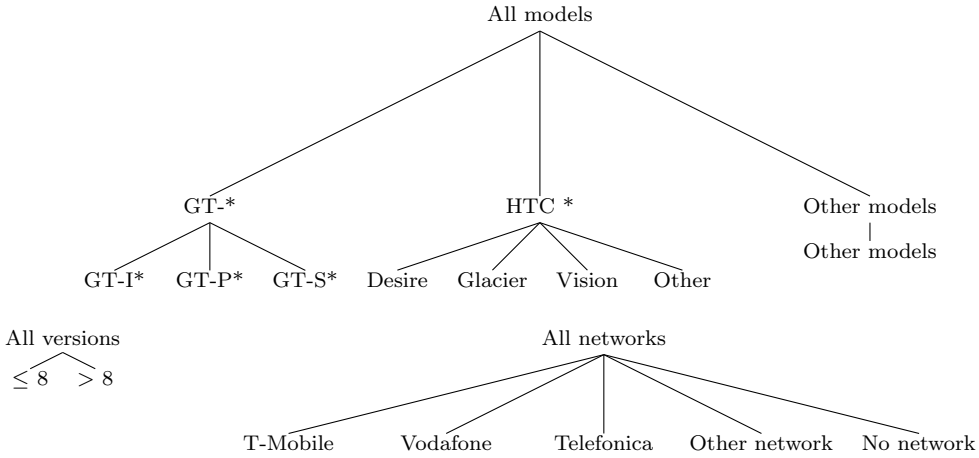

Figure 2: Value generalization hierarchies

their application and how the UTD Anonymization Toolbox was used to anonymize the assessed data.

4.3.1 Generalization

To provide k-anonymity for a table generalization as described in chapter 2.2 can be used.

Identifying possible generalizations has to be done manually. As most attributes are strings, dropping some characters is a possible solution. Checking whether this provides a significant change an the anonymity sets can be done using the SQL-query in listing 5. This query counts how many tuples in S have a network that start with the same character.

```
SELECT count(*) as cnt,
  SUBSTR(network, 1, 1) AS net
FROM android_data
GROUP BY net ORDER BY cnt;
```

Listing 5: Generalization on attribute network

The result of this query shows that there are 9 tuples identified uniquely by the first character of the network provider. Not distinguishing between upper- and lowercase letters, this can be reduced to 6 tuples. For this microdata to be released, simply dropping characters is not enough, as no k-anonymity can be provided. Further grouping is needed here. The statement in listing 6 groups the tuples by ranges of networks with the same first sign. It returns 337 tuples with the first letter of network between A and H, 134 between I and P and 518 between Q and Z.

```
SELECT count(*) AS cnt,
  CASE WHEN UPPER(SUBSTR(network, 1, 1))
    <= "H" THEN "A–H"
  WHEN UPPER(SUBSTR(network, 1, 1))
    <= "P" THEN "I–P"
  ELSE "Q–Z" END AS net
FROM android_data GROUP BY net;
```

Listing 6: Further generalization on attribute network

This approach can be used for the attributes device, model and version as well as combinations of these attributes.

Assuming only version as quasi-identifier, the data would be already 2-anonymous. The SDK version 9 (Android 2.3 - Android 2.3.2) only appears two times in the data. All other SDK versions appear more often.

This extensive grouping can provide anonymity. However, the information value whether a network provider starts with a letter between A and H or I and P is very low. For researchers, data generalized in such a way is generally useless.

To generalize the data in a way useful for others, a different approach was chosen. The appearance of models, devices and networks in S was counted. Models, devices and networks that appeared often in the data were summarized to families. Less used values were grouped into an 'Other' category. How this generalization scheme was developed is precisely described in Kern's seminar paper [5].

Generalization of attributes can be shown as trees. Sweeney called such trees value generalization hierarchies [11]. Each child node is an ungeneralized value. Inner nodes combine ungeneralized or less generalized values. The root node is the furthest possible generalization. Figure 2 shows the value generalization hierarchies for the attributes model, version and network. The generalization tree for devices looks very similar to the one generalizing the model. A division into Samsung GT, HTC and other devices is reasonable as those were the most common devices in the study [5]. Due to the mass of different ungeneralized values, the trees do not include those.

4.3.2 Suppression

Furthermore than generalizing values, it is possible not to release the value of some attributes. This is called suppression [11]. Suppression can be achieved by adding one generalization level to a value generalization hierarchy. This level suppresses all information this attribute could reveal. How the attribute version of Wagner's data can be suppressed is shown in figure 3.

doi: 10.2313/NET-2013-08-1_06

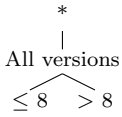

Figure 3: Value generalization hierarchy with suppression

4.3.3 *Anonymization with the UTD Toolbox*

The UTD Anonymization Toolbox was used with the generalization hierarchies proposed in chapter 4.3.1, on $F[model, device, version, network, longitude, latitude]$ and different suppression levels to create anonymized data. The number of tuples in which values might be suppressed is set to k. This is the standard setting of the UTD Anonymization Toolbox, as proposed by [11].

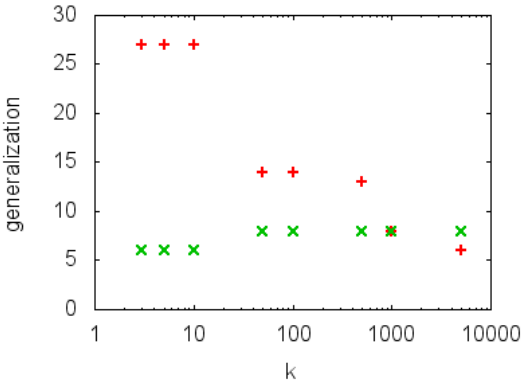

Figure 4: Number of different possible generalizations

The used algorithm looks for a combination of generalizations that alters the data as little as possible. Looking at the value generalization hierarchies, the used generalizations shall be as far down in the tree as possible. How many different possible combinations of the generalizations satisfy k-anonymity is shown by the red points in figure 4. The green points show how many generalizations have to be made for all attributes in total. A value of 6 means that the anonymization scheme with the lowest number of generalizations generalizes 6 times. For 3-anonymity the proposed generalization is {GT-I*, GT-P*, GT-S*}, {All devices}, {≤8, >8}, {T-Mobile, Vodafone, Telefonica, Other networks, No network}.

The UTD Anonymization Toolbox selects one anonymization scheme and outputs the data in anonymized form. An excerpt of the simplified database which satisfies 3-anonymity is shown in listing 7. Based on this data, the distribution of different android models or mobile phone carriers of smartphones can be assessed. Therefore the toolbox has successfully transformed the data in a 3-anonymous state.

```
(model, device, version, network,
    longitude, latitude)
('**********', 'GT-*******', '***', 'Other
    networks', 77.11, 28.39)
('**********', 'GT-*******', '***', 'Other
    networks', -47.53, -15.45)
('**********', 'GT-*******', '***', 'Other
    networks', -47.53, -15.45)
('**********', 'GT-*******', '***', 'Other
    networks', -47.53, -15.45)
('**********', 'GT-*******', '***', 'Other
    networks', 77.11, 28.39)
('**********', 'GT-*******', '***', 'T*-*
    Mobile', 15.01, 50.48)
('**********', 'GT-*******', '***', 'T*-*
    Mobile', 4.29, 51.55)
('**********', 'GT-*******', '***', 'T*-*
    Mobile', 5.37, 52.21)
('**********', 'GT-*******', '***', '
    Telefonica', -47.55, -15.46)
('**********', 'GT-*******', '***', '
    Telefonica', 2.31, 41.42)
('**********', 'GT-*******', '***', '
    Telefonica', -47.55, -15.46)
```

Listing 7: Excerpt of 3-anonymous table

Using SQL-querys similar to those presented in chapter 4.2 the resulting tables are checked to satisfy k-anonymity as configured in the settings file. In the 3-anonymous version, the attributes model and version are suppressed. For shortness, those are not shown in listing 8. This listing shows that all combinations of values for the quasi identifiers appear at least three times in the table.

```
(count, device, network)
(3, 'GT-*******', 'T*-*Mobile')
(4, 'saga/vision/glacier', 'Telefonica')
(12, 'Other devices', 'T*-*Mobile')
(12, 'saga/vision/glacier', 'No network')
(13, 'Other devices', 'Telefonica')
(23, 'GT-*******', 'No network')
(25, 'Other devices', 'Vodafone***')
(54, 'saga/vision/glacier', 'Other networks
    ')
(91, 'saga/vision/glacier', 'Vodafone***')
(95, 'GT-*******', 'Telefonica')
(112, 'GT-*******', 'Vodafone***')
(126, 'GT-*******', 'Other networks')
(127, 'saga/vision/glacier', 'T*-*Mobile')
(133, 'Other devices', 'Other networks')
(159, 'Other devices', 'No network')
```

Listing 8: Generalized quasi identifiers in 3-anonymous table

5. RELATED WORK

In his master seminar paper [9], Sebald discusses the impact of k-anonymity for researchers. He describes how AOL, Netflix and GIC have released data for research. This data seemed to be anonymous, but reporters for the New York times and professors from Texas University were able to identify users by linking attacks. Sebald does not present any evaluation of data on his own.

Users providing data can never be sure that the recipient handles their data with sufficient care. For the use of location based services there is need for users to access those services without disclosing their location. In his student research project, Greschbach proposes the use of realistic dummies for anonymization [3]. A user sends several requests with different locations when using a service. He

doi: 10.2313/NET-2013-08-1_06

will receive different answers, one for each request. Then the appropriate response needs to be selected. The location based service will never know the exact position of the user as he cannot determine the correct location from the set of requests. For a single user this provides anonymization. The network load increases due to the need of sending several requests and responses for each action. For the data holder this does not mean that there is no more need for anonymization. If one user does not know how to set up his device for such anonymization, the overall data does not satisfy any anonymity level.

In his seminar paper Kern assessed the same android data from this paper with regard to l-diversity [5]. He uses k-anonymity as base and shows which attacks cannot be held off using k-anonymity. He describes l-diversity and how it can be used as defense against those attacks. Using the anonymity evaluation of Wagner's data [15] in this work, he shows how a suitable value generalization hierarchy is developed. The presented hierarchy is used for anonymization in this paper. Similar to the use of the UTD Anonymization Toolbox in this paper, Kern uses the toolbox to provide l-diversity for the data.

6. CONCLUSION

Based on examples the need for anonymous microdata has been shown. Combination of insensitive attributes can be used to link different tables. Attackers can relate tuples of a study to single people. K-anonymity uses generalization and suppression to achieve anonymity. In a k-anonymous table each combination of values for a quasi identifier has to appear at least k times. This ensures that a single tuple is indistinguishable from at least $k-1$ different tuples based on the quasi identifier.

K-anonymity provides a level of anonymity that can be easily achieved. There are several algorithms to convert raw data into a k-anonymized form. Toolboxes can help researchers to anonymize their data before it gets released. If stronger anonymity is needed, l-diversity and t-closeness are possible solutions.

The data collected for [15] from android devices in its raw form provides no anonymity, although identifiers like G-mail account are not collected. If an attacker knows which tuples belong to the same device, it is possible to identify a range of single users by looking only at the device name or network provider. The first letter of the network is still enough to identify 9 out of 989 users. If it is not possible to distinguish between tuples of the same or different devices, the anonymity level is higher. Nevertheless it is possible to identify 30 users based on device, model, version and network.

To bring this data in k-anonymous form a value generalization hierarchy has to be created. For the android data, grouping into different device and network families is reasonable. With an existing value generalization hierarchy, the UTD Anonymization Toolbox can be used to create k-anonymous versions of the data.

7. REFERENCES

[1] Bundesdatenschutzgesetz §3. `http://www.gesetze-im-internet.de/bdsg_1990/__3.html`, Dec. 1990. [Online; accessed 12-December-2011].

[2] D. E. Denning and T. F. Lunt. A multilevel relational data model. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 220–234, Oakland, 1987.

[3] B. Greschbach. *Location Privacy l-diversity durch realistische Dummies*. Studienarbeit, Albert-Ludwigs-Universität Freiburg.

[4] A. Kemper and A. Eickler. Das relationale Modell. In *Datenbanksysteme Eine Einführung*, chapter 3. Oldenbourg Wissenschaftsverlag GmbH, München, 7 edition, 2009.

[5] M. Kern. *Anonymity: Formalisation of Privacy - l-diversity*. Seminar paper, Technische Universität München, Apr. 2013.

[6] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 49–60, 2005.

[7] N. Li and T. Li. t-closeness: Privacy beyond k-anonymity and l-diversity. *International Conference on Data Engineering (ICDE)*, (2), 2007.

[8] A. Machanavajjhala and D. Kifer. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, 1, 2007.

[9] S. Sebald. *k-Anonymity und dessen Einfluss auf die Forschung*. Seminararbeit, Albert-Ludwigs-Universität, Freiburg, 2010.

[10] Statistic Netherlands, Statistics Canada, Germany FSO, and University of Manchester. Glossary of Statistical Terms. `http://stats.oecd.org/glossary/detail.asp?ID=6961`, 2005. [Online; Accessed 16-February-2013].

[11] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):571–588, 2002.

[12] L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, Oct. 2002.

[13] J. D. Ullman. *Principles of Database and Knowledge-Base Systems (Volume I)*. Computer Science Press, Rockville, 1988.

[14] UTD Data Security and Privacy Lab. UT Dallas Anonymization Toolbox. `http://cs.utdallas.edu/dspl/cgi-bin/toolbox/anonManual.pdf`, Feb. 2010. [Online; Accessed 14-March-2013].

[15] S. Wagner. *User-assisted analysis of cellular network structures*. Bachelor's thesis, Technische Universität München, 2011.

[16] G. Wang. Cornell Anonymization Toolkit. `http://sourceforge.net/projects/anony-toolkit/files/Documents/cat-mannual-1.0.PDF/download`, May 2011. [Online; Accessed 14-March-2013].

[17] X. Xiao, G. Wang, and J. Gehrke. Interactive anonymization of sensitive data. *Proceedings of the 35th SIGMOD international conference on Management of data*, pages 1051–1054, 2009.