# Sharing data for public security

Michele Bezzi, Gilles Montagnon, Vincent Salzgeber, and Slim Trabelsi

SAP Labs, France,
name.lastname@sap.com

**Abstract.** Data sharing is a valuable tool for improving security. It allows integrating information from multiple sources to better identify and respond to global security threats. On the other side, sharing of data is limited by privacy and confidentiality. A possible solution is removing or obfuscating part of the data before release (anonymization), and, to this scope, various masking algorithms have been proposed. However, finding the right balance between privacy and the quality of data is often difficult, and it needs a fine calibration of the anonymization process. It includes choosing the 'best' set of masking algorithms and an estimation of the risk in releasing the data. Both these processes are rather complex, especially for non-expert users. In this paper, we illustrate the typical issues in the anonymization process, and introduce a tool for assisting the user in the choice of the set of masking transformations. We also propose a caching system to speed up this process over multiple runs on similar datasets. Although, the current version has limited functionalities, and more extensive testing is needed, it is a first step in the direction of developing a user-friendly support tool for anonymization.

## 1 Introduction

Governmental agencies, corporates, academic and research institutions hold a huge amount of data containing information on individual people or other sensitive data. They have often to release part of these data for research purposes, data analysis or application testing. For example, sharing of log data has been proved a valuable resource for research in network security against coordinated attacks [9], and collecting these data from multiple organizations allow to analyze the emergence of worldwide threats.

However, these data contain sensitive information and organizations are hesitant to share them. To reduce the risk, data holders use masking techniques (*anonymization*) for limiting disclosure risk in releasing sensitive datasets, such as generalizing the data, i.e., recoding variables into broader classes (e.g., releasing only the first two digits of the zip code or removing the last octet of an IP address) or rounding numerical data, suppressing part of or entire records, randomly swapping some fields among original data records, permutations or perturbative masking, i.e., adding random noise to numerical data values.

These anonymization methods increases protection, lowering the disclosure risk, but, clearly, they also decrease the quality of the data and hence its utility [4]. Finding the ideal balance between risk and utility and identifying the

right set of anonymization methods, among the many possible ones, to reach this equilibrium point is the main challenge of the data masking process. To this scope, there is the need to derive some criteria to assist the user in the choice of the set of transformations to be applied. In particular, we need to set the context specific requirements that define the information that has to be preserved, and use suitable metrics to quantify the disclosure risk in releasing the data. To address the latter point, various metrics for estimating disclosure risk have been proposed so far [3, 12, 1, 7]. They are typically based on the following attack scenario: an attacker has the knowledge about some variables, which may identify a record in the dataset. Considering the example of a medical database, the attacker may know a few attributes (age, gender, marital status) from an external public register (e.g., census data) or other source of information (e.g., knowing age and address of his neighbor). He then tries to match these variables (*keys*) with the partly altered records in the released database. In case of stochastic masking transformations, this matching may use probabilistic algorithms [14, 3, 8, **?**]. In the case of log files, an attacker may inject some information (e.g., scanning some specific ports), with the goal of later recognizing them in the anonymized logs. When a unique record matches a combination of key variables, the attacker can re-identify the masked record, assuming he is certain that the record is in the dataset. Risk metrics quantify 'how difficult' is this process of re-identification.

Ideally, such metrics/criteria should help the user to choose the appropriate set of masking methods for a specific dataset.

The goal of this paper is two fold: First, we introduce the main challenges for anonymizing data, describing two possible scenarios where data sharing may be valuable (see Sects. 2.1 and 2.2) and outlining the general requirements for the anonymization process (see Sect. 2.3). Second, we propose a model for supporting the user in the anonymization process (see Sect. 3), which includes a disclosure risk estimator and an efficient method for searching the 'best' set of anonymization methods. In Sect. 4 we will describe a prototype implementation of this model. Finally, conclusions are drawn in the last section.

## 2 Use cases and challenges

To illustrate the problem let consider two possible scenarios: data sharing of log files, and data sharing of personal identifiable information (PII).

### 2.1 Sharing Network Logs

Computer attacks are becoming more coordinated and addressing multiple targets at the same time, with large number of compromised hosts from many different organizations, possibly, in different countries. Detecting and reacting to these attacks may require cooperation of many institutions, and, often, a large scale analysis of network log data from all the possible targets [9]. However, organizations are often reluctant to share data, because they fear the risk

of leaking sensitive information, or for privacy concerns or for not revealing the structure of their internal network, which may reveal potential weaknesses to further attackers. Consequently, to promote data sharing we need to deal with possible privacy and security concerns of data holders. To this scope, the idea is to remove or modify potential sensitive information before release using various data masking transformations . These transformations include recoding variables into classes (e.g., releasing only the last bytes of an IP address or considering just two classes for the port number), suppressing part of or entire records (also known as black marker [10]), randomly swapping some fields among original data records, one-to-one mapping on a defined random set of IP numbers or perturbative masking, i.e., adding random noise to the number of packets transmitted [5]. Clearly, in this process we have to preserve the relevant information for data analysis, thus the set of transformations used has to be calibrated to specific analytics method to be applied.

### 2.2 Sharing PII for improving public safety

Data sharing between public agencies, and public and private organizations, can help improving public safety. For example, sharing health-care data can improve scientific research, and enable early detection of disease outbreak, as shown by the Real-Time Outbreak Detection System [13], which is a syndromic surveillance system based on health data integrated with data collected routinely for other purposes, such as absenteeism data, sales of over-the-counter health care products, etc ... . Similarly, police and fire departments could integrate multiple data sources, and share their information to optimize their capability of providing a coordinated defense.

The continuous growth of digital data may ulteriorly boost these approaches, but the the same time it raises privacy issues, and contrast with the increasing citizen awareness on privacy, and permission to use personal data is often difficult to obtain without guareenting some privacy protection. Accordingly, it is becoming crucial to develop technical methodologies to allow sharing of data without losing privacy. As in the previous example on network log data, anonymization techniques may be used to remove or obfuscate the more privacy-risky information, enabling the collection of large datasets of heterogeneous data from multiple sources.

### 2.3 Challenges

The major challenges in anonymization are not related to develop novel masking methods, but more to use them in a effective ways in the different contexts. In other words, there is a number of anonymization algorithms, the issue is which of them to choose to perform the anonymization balancing the conflicting privacy and utility requirements.

Utility requirements typically express what the data consumer wants to preserve. They are clearly dependent on the specific application, in particular to define what fields have to be anonymized, and how much information should

be preserved. Still, some general requirements in increasing level of complexity include:

- Preserving syntax/format. A basic requirement: the syntax should be conserved. It implies that the syntactical rule for each attribute must be considered. E.g., IP addresses conventions, first vs. last digits in zip codes or credit card numbers.
- Preserving semantics. In some cases, there is the need to keep the 'meaning' of some attributes. Therefore, names should be replaced with meaningful names (possibly language-specific), diseases with diseases, .... To this scope, it may be needed, first, to have the necessary semantic information in the original dataset, then to have available databases with list of candidates for replacement.
- Preserving Relationships. Data themselves are often used as keys in relational database. In particular, unique identifiers, as Social security number, may play this role. Accordingly, in some cases, the anonymization process should mask this data in a consistent way, to avoid to lose the relationships between tables, for example hashing these values.
- Preserving the distribution of original data. E.g., the percentage of empty fields, or the distribution of diseases. This can be particularly relevant for heavy-tailed distributions, where extreme values have to be correctly sampled.
- Preserving consistency. Attributes are often correlated, so the anonymization process should be applied in a consistent way across multiple attributes. E.g., city, states, telephone numbers.

Privacy requirements are also strongly context dependent, in some cases privacy regulations impose specific constraints on the anonymization process (e.g., HIPAA safe-harbor rules for medical data), but individuals or organizations may define additional requirements.

Quantifying the privacy level is very important for all the above mentioned scenarios, it provides a metrics that supports data holder in gauging privacy risk-utility. Even if various measures have been proposed so far, they are still limited used in the real-world applications. Typical issues include:

- Performance. Most of the algorithms used for estimating privacy risk do not scale when huge amount of real data are used. E.g., shopping data can easily have the size of several gigabytes. For such application current privacy metrics are still too time consuming.
- Attacker model. To perform an estimation of the disclosure risk, we need to define the attack model, and its basic assumptions. Typically, it is assumed that a possible attacker may use some external source of information (dictionary) to match some fields (keys) in the anonymized dataset, and infer some other information (e.g., identity or the value of some attribute) that were hidden during the anonymization process. This raises various issues:
  - Definition of keys. Identifying which attributes, or combination of attributes, may be used for re-identification is sometimes a difficult task.


**Fig. 1.** The flow of the anonymization tool. Data are loaded from a database, then user selects the anonymization methods via a GUI or loading a policy file. The anonymization engine applies the masking transformation and computes the risk. If risk exceeds the the threshold $r_{MAX}$, the risk optimization module proposes additional masking, otherwise the anonymized dataset is exported.

  - Definition/access to *dictionaries*. The basic idea in many privacy metrics definition, is trying to link the anonymized data to some external (not anonymized) data source (dictionary). Such dictionaries are often difficult to identify and access for the data holder before the data are actually released. For example, the test data for the Netflix context has been, partially, de-anonymized using a different, not-expected, source of external information [6].
- Complexity. The impact of different masking transformations on the risk value is often difficult to assess, especially for not-expert user, since risk value may depend on the amount and content of data, the assumptions of the risk metrics, etc... . Accordingly, for the user it is often hard to select the optimal process to minimize the risk.

To address some of the issues above, we developed a tool that supports the user in the anonymization process. The protection model we propose here is composed by two core components (see Fig. 1):

- an Anonymization Engine, comprising a set of masking algorithms for anonymizing the original dataset and a disclosure risk estimator.
- a Risk Optimizer, which suggests the user the "best" combinations of masking transformations to decrease the risk under a pre-defined level.

## 3 An Anonymization tool

These components are integrated in the following process: a user wants to share a dataset, e.g., for data analysis purpose, but he also wants to minimize the risk to reveal sensitive or private information. To this scope, the original dataset has to be anonymized before release. The user loads the original dataset, and sets the level of disclosure risk he wants to attain ($r_{MAX}$) using a specific metric. The user can choose a first set of masking transformations, for example in the case of datasets containing personal information, remove the social security number, generalize the postal code and age, etc. . This step can be performed by the user via a suitable user interface (see Fig. 2(Top)), or loading a predefined *anonymization policy* written by some security expert in a machine-readable language (e.g., XML [10]).

The anonymization engine applies the masking algorithms as specified by the user or by the anonymization policy and estimates the disclosure risk using a risk metrics $r_i$ (e.g., using one of the metrics listed in Sect. 1). If the computed risk $r_i$ is lower than the maximum acceptable risk $r_{MAX}$, the anonymized dataset can be released. Otherwise extra masking steps are needed. In the latter case, the user may decide to mask additional fields in the dataset or additionally downgrading data in already masked field (e.g., editing the policy or using a graphical interface to select the additional transformations). This manual work is tedious and it needs a technical understanding of the effect of the various transformations on the risk value. To optimize this process, we developed the Risk Optimizer module, which performs a search on the space of possible masking transformations, estimate the corresponding risks and, then, proposes to the user the 'closest' ones to the original transformations, which do not exceed the maximum risk value. This search space is highly-dimensional, even if we limit the masking transformations to the suppression of fields (or, equivalently, the replacement with a random value), the number of possible states to explore grows exponentially with the number of fields and the number of records in the dataset, making unfeasible to run even local search in case of large datasets. However, in many applications, the anonymization process is run on multiple instances on the same type of data-sets, so we propose a bootstrapping approach for speeding up the search. The idea is running an exhaustive search on a reduced set of records and caching the corresponding transformation set/risk values in a lookup table. The test sample can be the dataset used in the first run or, if it is too large, a random sample of it. In the following runs, the Risk Optimizer module uses the lookup table to estimate the set of transformations that can be applied to reach the risk threshold chosen by the user, and, at the same time, it is the closest to those originally selected, and proposes it to the user. The user selects one

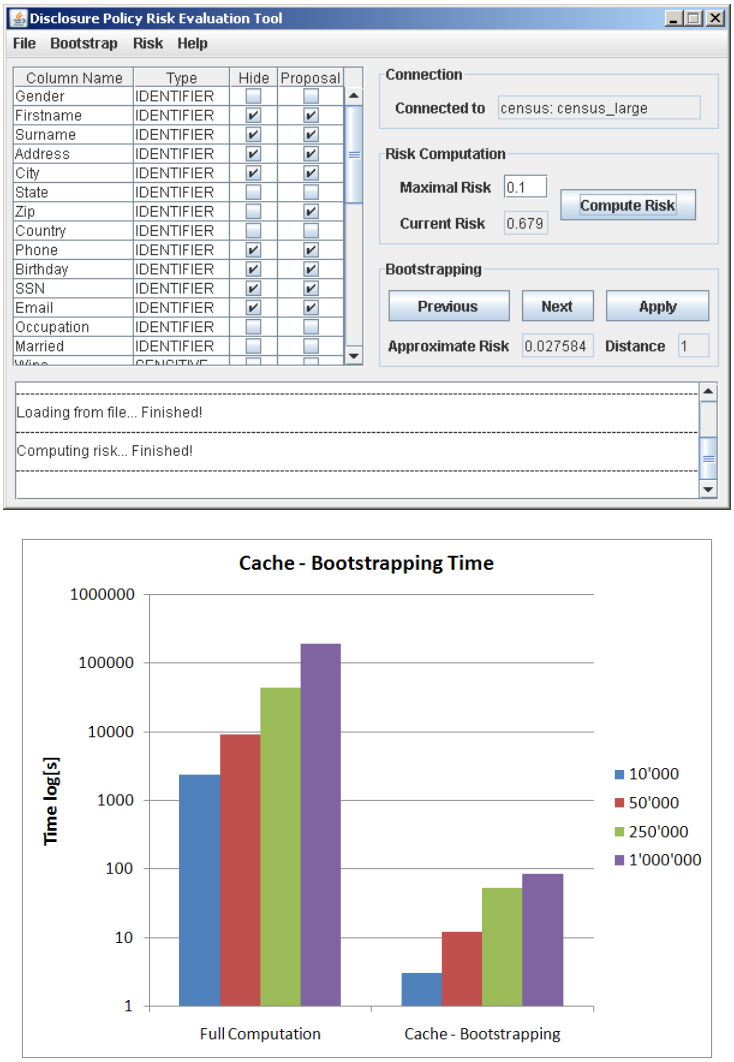


**Fig. 2.** Top: The Graphical User Interface. Ticking the 'Hide' box, the user can suppress or replace with random values the corresponding column in the dataset. After the first run, the Risk Optimizer suggests to suppress an additional column (the zip code in this case, see 'Proposal' column) to get to the maximum risk value allowed, as set by the user. Bottom: The runtime for using an exhaustive search (left bars) and the caching algorithm (right bars) for 4 datasets containing $10^4, 5 \times 10^4, 2.5 \times 10^5, 10^6$ records. A first dataset has been created for initialize the lookup tables. Then, using new datasets, we compute the processing time with/without using the caching system.

of them, and runs the Anonymization Engine to produce the masked dataset and, then, it checks to actual value of the risk. Clearly, if we change the type of dataset (e.g., medical data to log file), new lookup tables have to be created and stored.

We will show in the next section, how this approach can largely speed up the search process.

## 4 The prototype

We implemented the anonymization tool described above[1] . The tool has a graphical user interface, see Fig. 2(Top), which allows the user to easily load the dataset by querying a SQL database or fetching data from an SAP system, then he can create manually a disclosure policy by clicking on the attributes to display and the attributes to hide, or to load a predefined XML policy file. The current version supports only one kind of masking, i.e., suppression, so the user can simply decide to remove or not a certain field (column), clicking the tick box as shown in Fig. 2(Top) or by the appropriate policy file. User can also set the maximum value for the disclosure risk that he wants to achieve at the end of the anonymization.

By clicking on the 'Compute Risk' button the tool applies the chosen masking transformation and computes the corresponding final risk. The risk value is computed, first estimating the probability of re-identifying a single record, and then deriving the total percentage of records that could be re-identified (see Refs. [2, 11] for details).

If the risk exceeds the set threshold value, the user can manually set additional masking using the interface or ask to the system to search for a policy that matches the desirable risk. In this case, the bootstrapping system will propose to the user several less risky combinations that are close to his initial disclosure preferences, see Fig. 2(Top).

In Fig. 2(Bottom) we show the performance of the caching system for different size of the datasets. The caching system gives a large improvement in the performance of the system (note, the log scale for the processing time), fluctuations over multiple runs (not shown) are of the order of few percents. The testing datasets contain typical PII data: street address, city, zip code, country, gender, age, etc ... . They were randomly generated using a personal information generator [2].

## 5 Conclusions

Data sharing is a valuable tool for improving security, but privacy and confidentiality concerns restricts the sharing of data. Data anonymization is used

[1] For a detailed description of a first version of this prototype see Ref. [11]

[2] Fake Name Generator - http://www.fakenamegenerator.com/. This generator provides fake personal information with a realistic and coherent semantic meaning (e.g., valid city, state, and zip code combinations).

to address these issues, and various masking techniques are available. However, finding the ideal trade off between privacy and utility of the data is often difficult. Quantitative estimation of the privacy risk, privacy metrics, supports the user in the selection of the best combination of anonymization transformation, but the available metrics are typically computationally intensive, and they had a limited application to real-world scenarios up to now. In addition, non-expert user may find difficult to assess the impact of the different anonymization algorithms on the risk value.

In this paper, we illustrated the typical issues in the anonymization process, and presented a tool for assisting the user in the choice of set of masking transformations. This tool makes easy for not-expert user to select the minimal set of anonymization methods to reach a certain level of privacy risk, addressing one of the main difficulties of the anonymization process, that is the complexity of usage.

We also introduced a caching system to speed up this process over multiple runs on similar datasets. This allows to improve the performance of the tool in most of the application scenarios, addressing the performance requirement (see Sect. 2). Although, the presented model was tested on simple test data and includes only a small set of transformations, our preliminary results show that the tool, after an initial bootstrapping, can handle a dataset with one million of records in a rather short time. Clearly, introducing new masking transformations, such as generalization, will largely increase the dimensionality of the search space and introduce new challenges in terms of preserving the semantics of the data and the relationships between attributes. This case is currently under investigation.

## 6 Acknowledgements

The research leading to these results has received funding from the European Communitys Seventh Framework Programme (FP7/2007 2013) under grant agreement No. 216483.

## References

1. Benedetti, R., Franconi, L.: Statistical and technological solutions for controlled data dissemination. Pre-proceedings of New Techniques and Technologies for Statistics 1, 225–232 (1998)
2. Bezzi, M.: An entropy-based method for measuring anonymity. In: Proceedings of the IEEE/CreateNet SECOVAL Workshop on the Value of Security through Collaboration. Nice, France (September 2007)
3. Duncan, G., Lambert, D.: The risk of disclosure for microdata. Journal of Business & Economic Statistics 7, 207 (xx 1989), http://dx.doi.org/10.2307/1391438, 10.2307/1391438
4. Duncan, G., Keller-McNulty, S., Stokes, S.: Disclosure risk versus data utility: The RU confidentiality map. Technical paper, Los Alamos National Laboratory, Los Alamos, NM (2001)

5. Kounine, A., Bezzi, M.: Assessing disclosure risk in anonymized datasets. In: Proceedings of the FloCon Workshop (January 2009)
6. Narayanan, A., Shmatikov, V.: How to break anonymity of the netflix prize dataset (Oct 2006), http://arxiv.org/abs/cs/0610105
7. Samarati, P.: Protecting respondents' identities in microdata release. IEEE Trans. on Knowl. and Data Eng. 13(6), 1010–1027 (2001)
8. Skinner, C.J., Elliot, M.J.: A measure of disclosure risk for microdata. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64(4), 855–867 (2002), http://www.blackwell-synergy.com/doi/abs/10.1111/1467-9868.00365
9. Slagell, A., Yurcik, W.: Sharing computer network logs for security and privacy: A motivation for new methodologies of anonymization (2005), citeseer.ist.psu.edu/slagell05sharing.html
10. Slagell, A.J., Lakkaraju, K., Luo, K.: Flaim: A multi-level anonymization framework for computer and network logs. In: LISA. pp. 63–77. USENIX (2006)
11. Trabelsi, S., Salzgeber, V., Bezzi, M., Montagnon, G.: Data disclosure risk evaluation. In: Risks and Security of Internet and Systems (CRiSIS), 2009 Fourth International Conference on. pp. 35 –72 (oct 2009)
12. Truta, T.M., Fotouhi, F., Barth-Jones, D.: Assessing global disclosure risk in masked microdata. In: WPES '04: Proceedings of the 2004 ACM workshop on Privacy in the electronic society. pp. 85–93. ACM Press, New York, NY, USA (2004)
13. Tsui, F.C., Espino, J.U., Dato, V.M., Gesteland, P.H., Hutman, J., Wagner, M.M.: Technical Description of RODS: A Real-time Public Health Surveillance System. J Am Med Inform Assoc 10(5), 399–408 (2003), http://www.jamia.org/cgi/content/abstract/10/5/399
14. Yancey, W.E., Winkler, W.E., Creecy, R.H.: Disclosure risk assessment in perturbative microdata protection. In: Domingo-Ferrer, J. (ed.) Inference Control in Statistical Databases. Lecture Notes in Computer Science, vol. 2316, pp. 135–152. Springer (2002)