

LiDSec: A Lightweight Pseudonymization Approach for Textual Personal Information

Reza Rawassizadeh*, Johannes Heurix[†], Soheil Khosravipour*, A Min Tjoa*

**Institute of Software Technology and Interactive Systems*

Vienna University of Technology, Vienna, Austria

Email: rrawassizadeh@acm.org, skhosravipour@acm.org, amin@ifs.twien.ac.at

[†]Secure Business Austria

Email: jheurix@sba-research.org

Abstract—Sharing personal information benefits both users and third parties in many ways. Recent advances in sensor networks and personal archives enable users to record all of their digital objects including emails, social networking activities, or life events (life logging). These information objects are usually privacy sensitive and thus need to be protected adequately when being shared. In this work, we present a lightweight pseudonymization framework which allows users to benefit from sharing their personal information while still preserving their privacy. Tools like these are expected to be even more relevant in the near future, as personal information valuable for third parties is going to be more privacy sensitive than ever. Privacy-preserving data sharing increases the data owners' awareness of what they are actually sharing, thus rendering third party access more transparent.

Keywords-Pseudonymization; Personal Information; Privacy; Security

I. INTRODUCTION

In today's open world, our lives are centered on a habit that is epitomized by the Internet: information sharing. The Internet with its vast amounts of resources provides the perfect tool for sharing information, both on a professional as well as on a private level: Between business partners, traditional postal traffic has been more and more displaced by e-mails and thus communication has been considerably sped up, while social networks are more popular than ever and provide a sophisticated yet simple to use platform to inform friends and others of latest personal news and activities. This trend of data sharing with the society is expected to be increasing even more (cf. [5]). But all these facilitated information exchange opportunities come at a significant price: the sacrifice of privacy.

The availability of personal information whets the appetite of numerous companies that try to profit from this, especially for marketing purposes. A whole new branch of industry focuses on the collection of personal information such as visited web sites, movie preferences, and purchases over the Internet, using tracking cookies or the like to create individual profiles that are sold to third parties (cf. [1]). Technological advances, the increasing commercial profit of user profiling, and the rise in the amount of shared personal

and sensitive information - either wanted or unwanted - will very likely increase this problem in the near future even more.

To counter the privacy problem, several legal acts were introduced such as the Health Insurance Portability and Accountability Act (HIPAA) [25] that regulates the use and distribution of sensitive information to prevent fraud and data abuse. At the European level, the processing and movement of personal data is legally regulated by Directive 95/46/EC [9]. But often legal regulations are not that effective as they should be. For example, service providers that deal with personal information claim to preserve the user's privacy, but try to circumvent too restrictive regulations by using difficult-to-read terms and conditions to hide controversial passages and trapping the user into simply complying with all terms. Social security sites hide privacy-improving functions deep in the user interface such that they are easily overlooked (cf. [4]). Even if the service providers are reasonably trustworthy in protecting the individual's privacy, there remains the issue with insider attackers: Especially administrators with their extensive access rights are potential causes for critical data leakage incidents. Disgruntled employees may simply sell sensitive information to the highest bidder. Still, information sharing in general is highly beneficial, as long as the individual's privacy is not violated. The main point here is to let the user control with whom to share information and to what extent, leading to the requirement of a user-centered and user-controlled privacy solution.

In this work, we present LiDSec, a novel approach for privacy-preserving data distribution relying on pseudonymization, a technique which effectively hides the association between the information and the data owner (cf. [8]) and also reduces trust expectancies when sharing information in a community (cf. [13]). The main idea is to strictly control the information flow to someone instead of trusting this person not to abuse the sensitive information. This approach is owner-centered, i.e., the data owner specifically decides on which information to disclose by creating document-specific rules which

are processed to create non-critical datasets that can be published without compromising privacy. The proposed framework is sufficiently flexible to handle and process any type of text-based information¹ such as health records or life-logs.

The remainder of this work is organized as follows: In the next section, we provide background information on privacy-preserving techniques and related work. Then our privacy-preserving solution is presented in detail, followed by the description of a prototype as proof-of-concept. We then validate our concept by using a mobile life-log dataset as an example. Finally we summarize the benefits and any open limitations, and outline further work.

II. BACKGROUND

When sharing information, the challenge of ensuring privacy of personal information can be handled in different ways. The straightforward approach to limit unwanted data disclosure is to apply access control mechanisms. According to defined access rights (e.g. role-based, identity-based, etc.) managed in access control lists, persons are allowed to access only data elements they are cleared for. These access rights are usually managed by dedicated access control modules within information systems, or even by third parties, usually in combination with authentication mechanisms. For example, in [15], access to shared web content is restricted by the “circle of trust” of IM (instant messaging) networks. The proposed architecture requires that a particular user A first authenticates at an IM server which creates an access token depending on the user’s contact relationship to user B (e.g., friend, family, etc.) that can then be used to retrieve the desired resource from a web server. The problem with dedicated access control mechanisms is their single-point-of-failure: access control modules can be bypassed by, e.g., administrators with their unrestricted access rights (cf. [21]). Privacy-enhancing technologies (PETs) (cf. [10]) specifically deal with the privacy issue. Concepts such as unlinkability and anonymity do not rely on dedicated access control mechanisms to restrict access to sensitive information but apply depersonalization, disassociation, or encryption techniques to alter the data itself to limit unwanted information leakage. In [2], the authors add a cryptographic layer involving block-related hybrid encryption onto a client-server social networking architecture to hide the linkages between individual data content elements. Obviously, only those persons with access to the decryption keys are able to restore this content. In [3], attribute-based encryption (ABE) is applied to allow creating access “groups” by calculating so-called ABE secret keys (ASK) involving the set of attributes that define the particular groups. These approaches of privacy by data encryption require dedicated (and potentially complex)

¹We focus on structured text-based data that can be processed and consumed automatically (by, e.g., web services). Information represented in binary form (e.g., images) is outside the scope of this work.

key management schemes, and access revocations require the re-encryption and re-distribution of the new decryption key to the remaining persons, often a rather tedious process. The plain calculation overhead of cryptography may also be a hindering factor, especially when considering the limited processing power of mobile devices.

Anonymization relies on the fact that often only the association between specific data items needs to be broken up, while the individual items themselves can be stored in cleartext and present no privacy issue, containing only uncritical (e.g., publicly available) information. While anonymity can be achieved by depersonalization, i.e., the removal of person-identifying information, it may not be sufficient to simply remove any names. Often, there exist certain elements which are not identifying themselves, but may be so when in combination with other elements (e.g., combining age with ZIP code), depending on the level of semantic diversity within the dataset. These elements are denoted as quasi-identifiers in [22] where k-anonymity is achieved by generalizing and suppressing quasi-identifiers until each resulting combination cannot be distinctly assigned to one out of at least k individuals. The basic k-anonymity approach has been extended with l-diversity [14] and t-closeness [12] which address re-identification attacks exploiting background information or semantic similarities in datasets. A survey on further anonymization techniques, especially for social network data, can be found in [26].

Pseudonymization is a very similar technique to anonymization, but with the difference that person-identifying information is not completely removed but replaced with a pseudonym. This results in two distinctive advantages over plain anonymization:

- In contrast to anonymization, pseudonymization is reversible, if the need arises. While anonymization is obviously always accompanied by a certain loss of data accuracy and content, pseudonymization allows to restore the original dataset under strictly controlled circumstances. For example, pseudonymizing health information instead of fully anonymizing them for secondary use, i.e., research activities where initially the actual identities of the patients are not required to be known by the secondary users, allows to contact the patients for gathering further information, if the patients agree.
- The application of different pseudonyms allows to retain certain links of datasets and information. For example, using the same pseudonym for a set of forum posts links these posts to the same user (while still hiding the true identity of that particular user). However, using multiple pseudonyms for different threads hides the fact that it is indeed the same user.

Pseudonyms are usually applied for identity management (e.g., [6]): In [7], an anonymous credential system is in-

roduced that maps credentials to pseudonyms which allow limited tracking, and thus accountability. Therefore, transactions requiring user binding such as chargeable newspaper subscriptions can be realized while still hiding the actual individual’s identity. Still, global de-pseudonymization is possible by authorized parties in case of a credential abuse. Another application of pseudonymization is, as already indicated above, the area of e-health. Pseudonymization is usually conducted when making health records available for secondary use in research activities: The approaches in [18], [19] rely on a combination of hashing and encryption techniques to realize different pseudonymization scenarios, one-way (actually anonymization) and two-way (i.e., reversible) pseudonymization. In [24], pseudonymization is achieved by first separating the identification data from the anamnesis data which is then stored in a separate database referenced with so-called unique data identification codes (DIC) as pseudonyms. Another approach is presented in [16] that integrates primary and secondary usage of health data assigning so-called root pseudonyms only known to the patient and shared pseudonyms as authorization tokens for selected health care providers.

In contrast to these approaches, our pseudonymization solution is not limited to a particular application area but is able to handle any text-based data. We also do not rely on expensive cryptography, maximizing performance and efficiency for deployment on mobile and pervasive devices.

III. REQUIREMENTS

To achieve our goal of ensuring privacy-preserving data sharing, we have identified the following requirements:

- **User Control:** The most important requirement is to ensure user control, or more specifically, data owner control, i.e., let the data owner decide on the “amount” of privacy of published datasets. Unlike with typical privacy settings of, e.g., social networks, this also includes that the owner enforces her privacy demands in order to reduce the trust required in third parties. As already mentioned earlier in this article, one major threat is the existence of potential inside attackers, especially malicious administrators with extensive access rights that are able to circumvent user privacy settings.
- **Efficiency:** The straightforward (reversible) approach of preserving data confidentiality and privacy is to encrypt the sensitive elements. However, encryption requires dedicated key management and key protection strategies and thus is always associated with a certain amount of overhead, both storage- and performance-wise. While the execution of cryptographic operations is usually unproblematic on current conventional hardware (e.g., personal computers), the limited calculation power of pervasive devices such as mobile phones or tablet computers still pose a considerable obstacle for efficient use of key-based cryptography.

- **Adaptability:** Existing privacy-preserving solutions are usually aimed at specific application domains (especially e-health) and designed to handle certain types of information with a predefined data structure. Regarding the diversity of text-based data representation methodologies, an effective privacy-preserving solution is required to be highly adaptable to different data structures.
- **Reduction of Manual Work:** Considering the large amounts of information that are involved in data sharing these days, manually identifying and processing privacy-sensitive elements is usually quite tedious, if not infeasible. Therefore, reducing the manual work needed by users by automatizing as many tasks as possible is a major requirement in developing a viable privacy preservation solution. Still, user control has to be ensured at all times.

IV. LIGHTWEIGHT DATA SECURITY (LiDSEC) SYSTEM DEFINITION

In the following, we present LiDSEC, a pseudonymization approach for Lightweight Data Security. The system’s design was specified on the following considerations:

- Apply (selective) pseudonymization before publishing personal information.
- Let the data owner (in the following simply denoted as user) decide on which elements to pseudonymize by creating privacy rules depending on the document type.
- Then let the tool automatically screen for privacy-compromising elements within the documents to reduce manual pseudonymization work to a minimum.

The framework architecture (cf. Figure 1) is composed of six distinctive components: Data Adapter, Rule Settings, Converter Engine, Validator, and Reporting Module. The

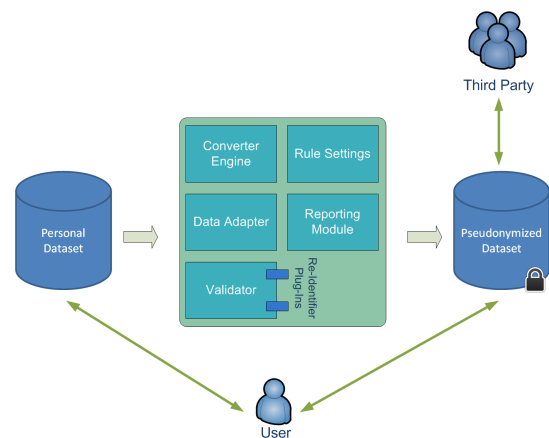


Figure 1. Conceptual Framework Architecture

Data Adapter is responsible for connecting to the source dataset and for extracting the data’s structure in an attribute-value style. Attribute-value pairs are denoted as information

objects. The records of a dataset should be encoded in standard formats such as XML or CSV. Depending on the actual data source, e.g., relational databases such as MySQL or simple JSON-encoded files, individual Data Adapters are required to be implemented by the framework users to manage the database connection details or file location handling. These can then be integrated with the pseudonymization framework to support multiple data sources at once. Our current implementation of LiDSec (cf. section V) uses a GUI wizard as shown in Figure 2, although the actual pseudonymization procedure can also be consumed as a service.

After connection to the dataset has been established or the location of the dataset been set (cf. screenshot 1 in Figure 2), the process of data structure identification is initiated. This process parses and recognizes entities and their sub-elements of the target dataset, based on the dataset’s format. As shown in Figure 2 (screenshot 3), the identified fields are presented to the user via the GUI in a structured way. If there are any non-recognized fields in the dataset, the user is thereby able to manually edit them. We refer to this as semi-automated data structure identification: First the framework implementation automatically goes through the dataset and searches for and extracts known information objects. Un-identified entities that are missed by the framework can then be added manually afterwards.

After the dataset’s structure has been identified, the user needs to decide on how to handle each entity (cf. Figure 2, screenshot 3). Basically there are four options:

- *Keep* the information object as it is.
- *Remove* the *value* of a particular information object, i.e., anonymize.
- *Remove* the complete *entity*, i.e., suppress the information object.
- *Change* or substitute the particular information object with a pseudonymized one.

Anonymization usually alters the overall semantics and accuracy of the original data and thus should be applied only when necessary. Similarly, suppressing entire information objects may be necessary on highly sensitive information objects where their plain existence may result in a privacy compromise. In general, pseudonymization is the better choice for preserving data expressiveness, but it still lies in the user’s hands to find an acceptable trade-off between usability of the sanitized dataset for the data consumer and the user’s privacy requirements.

The user’s choices are collected in the *Rule Settings* component of LiDSec which stores these choices as pseudonymization policies and applies them to the dataset during the pseudonymization process. In our current implementation of LiDSec, we store these policies in an external configuration file. These policies can be simply replicated for different ap-

plications and forwarded or modified to fit the needs of both the user and the data-consuming third party. For example, the user intends to consume a particular service from provider B and therefore needs to provide B with particular information. Using an existing policy set for publishing pseudonymized information to another data consumer A, both the user and B agree on a slightly modified policy that suits the automated processing facilities of the service provider, but still satisfies the user’s privacy requirements.

After the pseudonymization policies have been clarified and persisted in the configuration file, the *Converter Engine* initiates the actual pseudonymization process and converts or removes original information objects according to the pseudonymization policies to produce a sanitized dataset.

All replaced values, i.e., the mapping between original information and pseudonymized data (e.g. “John Smith” converted to “Person12”), are maintained in a map file. This map file can be reused if the user chooses to (as shown in Figure 2, screenshot 2) in order to ensure that each instance of a particular value is replaced with the same pseudonym previously used to preserve logic links. Alternatively, the user may choose to create a new file with new pseudonyms to semantically unlink the current pseudonymized dataset from any previous records containing the same values. Thereby, the user is able to carefully control the information flow concerning the linkage between multiple datasets by creating multiple virtual identities (using different pseudonyms) with individual sets of information.

Following the pseudonymization process, the outcome’s quality is checked by the *Validator*. We designed this Validator module to be easily extensible by external plugins to make use of existing re-identification approaches. In this context, we define re-identification as the process of successfully identifying the particular dataset’s individual by analyzing the pseudonymized dataset and residual personal information that should have been removed by de-personalization [20] or de-identification [23]. The result of this validation process is reported to the user by the *Reporting Module*.

V. PROTOTYPE IMPLEMENTATION

Our LiDSec prototype has been implemented in Java 7 with an optional GUI layer to assist users in working with the provided services; users can utilize the application by the GUI wizard or as a service. The prototype has been tested on a Mac Book Pro (MacOS 10.6) with a 2.4GHz CPU and 2GB memory, as well as on a Lenovo Thinkpad (Windows 7) with a 2.5GHz CPU and 4GB memory. No platform-specific library is required for the execution of the prototype. Configuration and map files are both encoded as XML files, which need to be kept safe by the user to prevent information leakage [11], [18].

In the following, we provide an example of a record (log) to be pseudonymized, containing a particular sms.

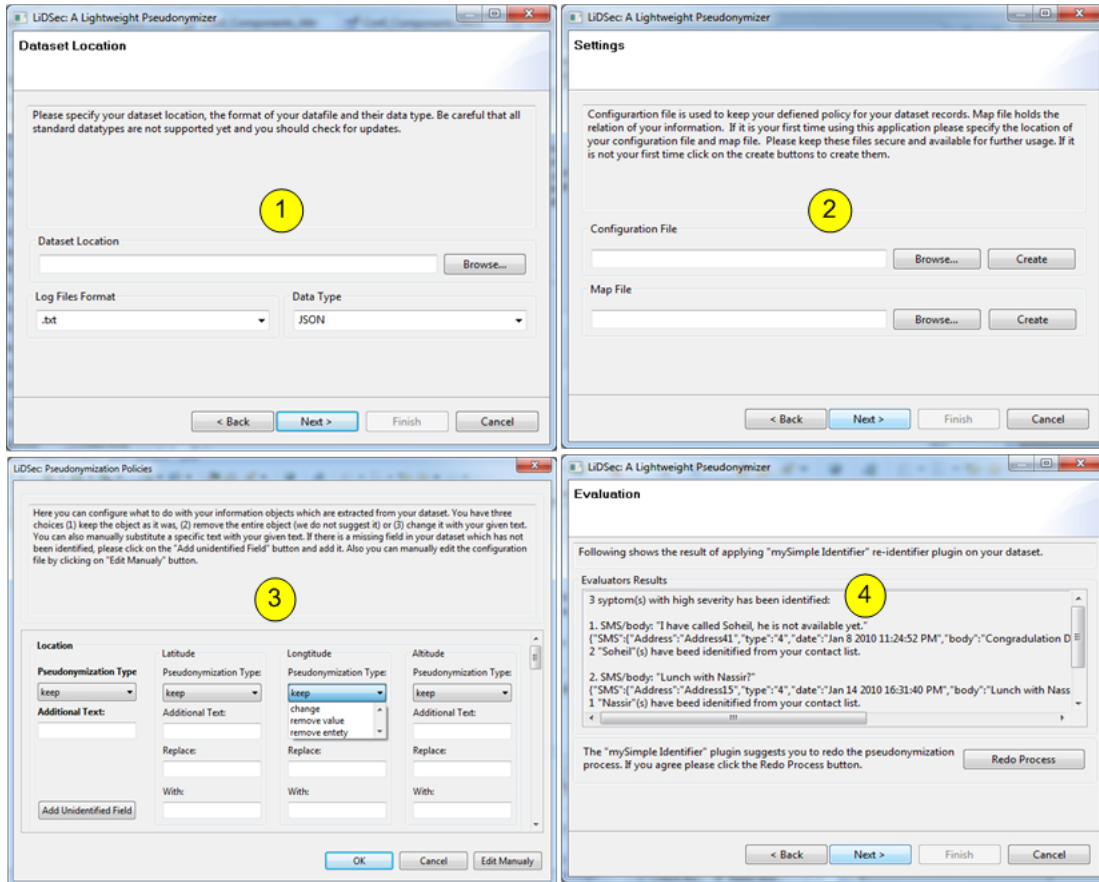


Figure 2. Screenshots from the Graphical User Interface

```
{
  "SMS": {
    "Address": "06802368296",
    "type": "1",
    "date-time": "Jan 14 2010 3:39:21 PM",
    "Body": "Plz Call me to schedule the gathering",
    "metadata": {
      "name": "John"
    }
  }
}
```

In this simple example, the user intends to publish this information to a data consumer who utilizes the data to analyze the frequency of location changes by different persons. The user regards the exact location and place as highly privacy sensitive and thus creates a pseudonymization policy that hides the exact location of the particular place. Furthermore, the metadata entry needs to be generalized. The resulting sanitized record is produced as follows:

```
{
  "SMS": {
    "Address": "address7",
    "type": "1",
    "date-time": "Jan 14 2010 3:39:21 PM",
    "Body": "body38",
    "metadata": {
      "name": "name"
    }
  }
}
```

The address and body are replaced with individual pseudonyms, while the metadata name’s value is generalized to the fixed value of “name”; the time stamp and provider entry are left as they were. By pseudonymization, the user now effectively hides the communication content and party, but still allows the third party to analyze at what time does the communication occurred.

VI. EVALUATION

We evaluated LiDsec from three different perspectives: pseudonymization effectiveness and quality, usability, and validity. Effectiveness is proven by testing the tool with real world data, while usability is evaluated by a user survey based on Nielsen’s usability heuristics [17]. Finally, validity is proven by revisiting the basic requirements stated in section III.

As our test case, we pseudonymized a personal life-log dataset composed of all activities a user performs with her mobile phone. The record actually consists of a set of log files - one log file per each day - encoded in JSON² format. These log files contain application usage, location changes, bluetooth device scan logs, received/sent text messages and received/dialed calls. To evaluate the quality of the resulting dataset, a simple re-identifier has been implemented as Validator component. Having access to the map file, this re-identifier scans the resulting pseudonymized dataset for any values stored in the map file, i.e., it checks whether a particular value of a specific attribute being pseudonymized according to a pseudonymization rule is

²<http://www.json.org>

still present within another non-pseudonymized attribute. For instance, it searches for “home” and reports text messages containing this string. Figure 2 (screenshot 4) shows a report where three re-identification symptoms are detected, all of them appearing in the text message content. In this example, the user pseudonymized the sender and receiver names, but not the (complete) text message content. As stated before, our simple re-identifier can be replaced (or extended) with more sophisticated approaches such as annotation and entity-recognition solutions. The user is then able to alter the pseudonymization policies and/or manually modify the record and remove these symptoms. In an iterative process, the policies are refined until no symptoms are identified any more.

The user interface’s usability is evaluated by five users of (relatively) similar age (between 23 and 31 years of age), but different gender (two women, three men) and computer expertise (four stated having above average computer knowledge and one having only basic knowledge). The representative task was to pseudonymize a single life-log dataset due to its simplicity and expressiveness containing human readable data. All participants received an introduction about the tool’s purpose, but no hints on how to actually use this tool. The results of the survey based on Nielsen’s usability heuristics indicated the highest score (5 of 5) in “visibility of the system status”, and the lowest score (1 of 5) in “help and documentation” as well as “recognition rather than recall”. Other heuristic factors received scores above average.

Finally, LiDSec meets the requirements of privacy-preserving data sharing as follows:

- **User Control:** Our approach revolves around the user to specify an individual pseudonymization policy set that can be either derived from a default set or created from scratch. This ensures the development of fine-grained and document type-specific privacy policies, including the option of replacing certain values with selected pseudonyms or removing these values, or even removing complete information objects to limit critical and unwanted information leakage when publishing privacy-sensitive data. The optional GUI supports the user in creating these policies.
- **Efficiency:** We developed a lightweight solution with the aim of being highly efficient. As it completely forgoes computationally-expensive cryptographic algorithms or other complex operations, it is suitable to be deployed on devices with limited computational capabilities (e.g. mobile devices). Our approach does not rely on specific third party solutions and is thus largely independent from software technologies. The actual pseudonymization process requires only simple string replacement operations and thus can be executed within a very small amount of time.
- **Adaptability:** The introduction of document type-

specific data adapters allows to adapt our solution to a whole range of document domains and potential data sources. Without a specific application area in mind, LiDSec is able to handle any text-based documents, as long as its structure is known. Potential data sources include simple text files, relational databases, or other data-providing services (e.g., web services). The prototype can be used as standalone tool with the GUI wizard as well as its functionality consumed as an intermediate pseudonymization service.

- **Reduction of Manual Work:** Data structure identification is handled in a semi-automated way, i.e., the document type and the involved entities are automatically determined by analyzing the document content and manual work only necessary when unknown entities are detected. Pseudonymization policies need to be created once by the data owner and are then re-used for all entities matching the rule sets. This semi-automated approach drastically reduces the required human workload for privacy-enabled data sharing.

While our solution is in general highly adaptable to any text-based information, we identified the following limitations due to our design decisions:

- As the targeted use is to publish machine-processable documents in a privacy-preserving manner, we focus on structured data elements only. Free-text documents cannot be supported by LiDSec.
- Data pseudonymization is executed only when publishing documents, and the original dataset needs to be kept at the data owner’s side. Therefore, our pseudonymization framework cannot act as data protection solution for locally stored documents.

VII. CONCLUSION

Personal information can be regarded as personal property. Users as data owners may provide third parties with these pieces of information and in return consume services provided by these parties. One particular issue with this kind of sharing of sensitive personal information is the compromise of the users’ privacy. In this work, we presented LiDSec, a solution that assists data owners in pseudonymizing their data according to their needs. LiDSec is designed to be largely independent from restricting software technologies and can be deployed on pervasive devices hosting privacy sensitive information. It is lightweight and highly adaptable to different data sources. Further work includes the extension of the tool with different data adapters and re-identifier plugins, as well as exploring its application on free-text data sources.

REFERENCES

- [1] The Web’s New Gold Mine: Your Secrets. <http://online.wsj.com/article/SB10001424052748703940904575395073512989404.html>, 2010. Last Access July 2010.

- [2] J. Anderson, C. Diaz, J. Bonneau, and F. Stajano. Privacy-enabling social networking over untrusted networks. In *Proceedings of the 2nd ACM workshop on Online social networks*, WOSN '09, pages 1–6, 2009.
- [3] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin. Persona: An Online Social Network with User-Defined Privacy. *SIGCOMM Computer Communication Review*, 39(4):135–146, 2009.
- [4] J. Bonneau and S. Preibusch. The Privacy Jungle: On the Market for Data Protection in Social Networks. In *Economics of Information Security and Privacy*, pages 121–167. Springer US, 2010.
- [5] J. Breslin and S. Decker. The Future of Social Networks on the Internet: The Need for Semantics. *IEEE Internet Computing*, pages 86–90, 2007.
- [6] J. Camenisch, a. shelat, D. Sommer, S. Fischer-Hübner, M. Hansen, H. Krasemann, G. Lacoste, R. Leenes, and J. Tseng. Privacy and Identity Management for Everyone. In *Proceedings of the 2005 Workshop on Digital Identity Management*, DIM '05, pages 20–27, 2005.
- [7] J. Camenisch and E. Van Herreweghen. Design and Implementation of the idemix Anonymous Credential System. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, CCS '02, pages 21–30, 2002.
- [8] D. Chaum. Security Without Identification: Transaction Systems to Make Big Brother Obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.
- [9] European Union. Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. *Official Journal of the European Communities*, L 281:31–50, 1995.
- [10] S. Fischer-Hübner. *IT-Security and Privacy: Design and Use of Privacy-Enhancing Security Mechanisms*. Springer, Berlin, 2001.
- [11] U. Flegel. Pseudonymizing Unix log files. In *International Conference on Infrastructure Security, InfraSec 2002*, pages 162–179, 2002.
- [12] N. Li, T. Li, and S. Venkatasubramanian. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In *IEEE 23rd International Conference on Data Engineering (ICDE2007)*, pages 106–115, 2007.
- [13] A. Lysyanskaya, R. Rivest, A. Sahai, and S. Wolf. Pseudonym Systems. In *Selected Areas in Cryptography*, pages 184–199, 2000.
- [14] Machanavajjhala, Ashwin and Kifer, Daniel and Gehrke, Johannes and Venkatasubramanian, Muthuramakrishnan. L-diversity: Privacy beyond k-anonymity. *ACM Transaction on Knowledge Discovery Data*, 1(1):3, 2007.
- [15] M. Mannan and P. C. van Oorschot. Privacy-enhanced Sharing of Personal Content on the Web. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 487–496, 2008.
- [16] T. Neubauer and J. Heurix. A Methodology for the Pseudonymization of Medical Data. *International Journal of Medical Informatics*, 2010.
- [17] J. Nielsen and R. Molich. Heuristic Evaluation of User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Empowering people*, pages 249–256, 1990.
- [18] K. Pommerening. Medical Requirements for Data Protection. In *Proceedings of IFIP Congress*, volume 2, pages 533–540, 1994.
- [19] K. Pommerening and M. Reng. Secondary Use of the EHR via Pseudonymisation. *Medical Care Computetics 1, IOS Press*, pages 441–446, 2004.
- [20] A. Rector, J. Rogers, A. Taweel, D. Ingram, D. Kalra, J. Milan, P. Singleton, R. Gaizauskas, M. Hepple, D. Scott, and R. Power. CLEF-Joining up Healthcare with Clinical and Post-Genomic Research. In *Proceedings of UK e-science all hands meeting*, pages 264–267, 2003.
- [21] R. Russell, D. Kaminsky, R. F. Puppy, J. Grand, D. Ahmad, H. Flynn, I. Dubrawsky, S. W. Manzuik, and R. Permeh. *Hack Proofing Your Network (Second Edition)*. Syngress Publishing, 2002.
- [22] P. Samarati and L. Sweeney. Protecting Privacy when Disclosing Information: k-anonymity and its Enforcement through Generalization and Suppression. Technical report, SRI Int'l, 1998.
- [23] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty Fuzziness and Knowledge Based Systems*, 10(5):557–570, 2002.
- [24] C. Thielscher, M. Gottfried, S. Umbreit, F. Boegner, J. Haack, and N. Schroeders. Patent: Data Processing System for Patient Data. *International Patent, WO 03/034294 A2*, 2005.
- [25] U.S. Congress. Health Insurance Portability and Accountability Act of 1996. *104th Congress*, 1996.
- [26] B. Zhou, J. Pei, and W. Luk. A Brief Survey on Anonymization Techniques for Privacy Preserving Publishing of Social Network Data. *ACM SIGKDD Explorations Newsletter*, 10:12–22, 2008.