Distributed Denial of Privacy

Hannes Grunert
Database Research Group
University of Rostock
Albert-Einstein-Straße 22
18051 Rostock, Germany
hg(at)informatik.uni-rostock.de

Abstract: For many years, privacy has not been considered as a critical issue in the field of software development. Recent events have led to discussions how the privacy of the user can be protected in information systems. While most approaches do only focus on increasing the security of the systems to protect private information against uncontrolled access, the aspect of enhancing privacy itself has been ignored. Our framework, called PArADISE (Privacy AwaRe Assistive Distributed Information System Environment), maximizes privacy by minimizing the amount of data stored and queried by information systems. In this paper, we present a new method to protect the anonymity of the user, even if multiple queries are combined.

1 Introduction

Assistance systems in smart environments are collecting a lot of data via different sensors (e.g. the SenseFloor project [SL08] and passive RFID tags [WT13]) in order to compute the intentions and forecast future activities of the user. In most cases the data contain more information than needed for achieving the purpose of the system.

The aim of our work is the privacy-aware propagation of the sensor- and context-information towards the analysis-tools of the assistance system. The privacy needs of the users are transformed into integrity constraints of the database system storing the collected information. Comparing the information needs and the privacy claims of the users, the privacy component of the assistance system selects, reduces, compresses or aggregates the data at the sensor as closely as possible.

Our approach is implemented in the PArADISE-Framework (Privacy-AwaRe Assistive Distributed Information System Environment) and is going to be evaluated in the Smart Appliances Lab ([EK05]) at the DFG-Graduate school MuSAMA (Multimodal Smart Appliance Ensembles for Mobile Applications).

Previous work has been limited to protect the privacy of the user in a single result set. By combining the results of multiple queries, an honest-but-curious adversary can gain access to secret information. In this paper, we propose a new methodology to observe queries using a sliding window. During the lifetime of this window, a user can not be identified.

The rest of this paper is organized as follows. In the second section the related work is

presented. The third section gives a brief overview of the superior topic. Issues regarding privacy concerns by connecting information are discussed in the next section. Some conclusions are drawn in the final section.

2 Related Work

For the compliance with privacy-protection different access-control-mechanisms and -algorithms which ensure the anonymity or pseudo-anonymity of personal records exist. Most algorithms base on k-anonymity ([Swe02]). It proposes a method that splits data into quasi identifier (QIs) and sensitive information (SIs). The parameter k shows, that k tuples are indistinguishable by using only the QIs. Thus, re-identification attacks are blocked.

Other techniques, such as l-diversity ([MKGV07]), t-closeness ([LLV07]), Data Slicing ([LLZM12]) and Bucketization ([KG06]) improve k-anonymity by preventing similarity attacks.

Unfortunately, one of the major drawbacks to these algorithms is that information can be combined to unreveal private data, even if it was highly anonymized before. In this paper, we present a new method to get rid out of this limitation.

3 PArADISE

In order to enhance privacy in assistance systems, we are currently developing a framework called PArADISE (Privacy AwaRe Assistive Distributed Information System Environment). In this prototype, the user can specify detailed privacy settings in a policy. The users privacy concerns are checked against the information need of the assistance system.

The aim is that the user offers the system enough data to operate as desired, but can not infer more information about him or her. For that task, we are currently implementing a privacy engine, which consists of the privacy policy and a privacy-aware query processor.

Our privacy model is based on the W3C draft for Privacy Preferences Platform (P3P, [W3C07]) and the Bell-LaPadula Security model ([BL73]). As an extension, we adopt continuative concepts for data selection and aggregation.

The access to information can be designed in a more fine-grained way by introducing additional gradations between locking and releasing data. Thus, both the information needs of the assistance system and the privacy rights of users can be satisfied. Possible intermediates can be obtained by selection, restriction, compression and aggregation, as well as by the insertion of the fuzzy values. For reasons of space, our privacy model is not addressed in this paper. For more information, please refer to [Gru14].

The privacy-aware query processor is composed of a pre-processor for incoming queries and a post-processor for the result set. Whereas the pre-processor modifies the query

to reduce the amount of data fetched from the data storage, the post-processor applies privacy-algorithm before transmitting the data back to the requester.

During the pre-processing stage, the preliminary query is analyzed and checked against the privacy policy of the concerned user. Every affected personal information queried by the system is monitored, whether it is uncovered by the user at all (projection) and if it can be used under user-defined constraints. This constraints can be used to decide if the revealed information are preselected or aggregated. Furthermore, it is also checked if the processing node has enough capacity (load balancing). Also, it is tested if the information system could gain enough information to produce satisfactory results. Finally, the constraints are used for a modification of the query which fulfills all needs. The reformulated query is executed on the data storage holding the personal information. The gained results are passed towards the postprocessor.

Taken the preliminary result from the intermediate stage, the post-processor checks information needs and privacy settings. This time, the result is modified with privacy-preserving algorithm like k-anonymity ([Swe02]) or data-slicing ([LLZM12]), if and only if the processing unit has enough power. Thus, the modified result is send back to the requester. A serious limitation with this approaches, however, is that they operate on a single result set. In the next chapter, we present a new method which uses a sliding window to monitor a data pool holding personal information.

4 Maintaining anonymity through multiple queries

The algorithms for achieving privacy presented in section 2 only work on single queries. The anonymization is insufficient, if a single user executes multiple queries on the database. While individual result sets for themselves can guarantee the anonymity of a user, the connection of the results can disturb privacy.

The information in the database form an universe U. The adversary do not know anything about U before he executes his first query q_1 on it. The outcome of query q_1 (after applying privacy-algorithm) is the set A. Executing a second query q_2 results in a set B. Now, the adversary has the information of both queries. Furthermore, he can connect set A and B ($A \cap B$) to get all tuples which belongs to both queries. Also, he knows which tuple appertain to set A but not to set B ($A \backslash B$), and which tuple appertain to set B but not to set A ($B \backslash A$). So, all three sets ($A \cap B, A \backslash B, B \backslash A$) have to checked against privacy-algorithm to guarantee anonymity.

Example A database db stores information about employees (name) and their income (salary). Query A sums the salary up (*SELECT SUM(salary) FROM db*), query B does the same, but without an employee named P (*SELECT SUM(salary) FROM db WHERE name != "P"*). Both queries provide only aggregated, statistical information. Nevertheless, it is obvious to see, that subtracting the result B from A reveals the salary of person P, which equals the subset $B \backslash A$.

When a database is executed over a long time, the number of queries ascends continuously. Thus, the number of subsets increases exponentially ($2^n - 1$ subsets for n queries). Guaranteeing anonymity on every subset confronts the system with two problems. First, the privacy-algorithm have to be applied on every non-empty subset. Having a huge amount of subsets leads to a long execute time, which effects the respond time of the system.

To solve these problems we propose the method shown in algorithm 4. As input, it catches a continuous stream of queries *Q*. The algorithm is executed for every new query of the system. Whilst the maximum length *n* of sliding window *W* is not exceeded, every query *q* is added to *W*.

Query *q* is analyzed and combined with the other queries of *B*. *q* is connected with a logical AND with the known queries of *B*. The same is done with the complement NOT(q). Furthermore, *q* is connected with every complement of *B*.

Thus, all subsets are stored in *B*. For every subset, an anonymization-algorithm (e.g. [Swe02]) is applied to monitor privacy claims.

If the maximum length is exceed, the oldest query *q_old* is removed from *W* and all occurrences of *q_old* are removed from *B* to realize the sliding window. Afterwards, the algorithm is repeated for the new query list.

In most cases, checking anonymity for queries from a single peer is insufficient. Experience from the field of Web Security shows that attacks can also been distributed on multiple peers. Traditional kinds of Distributed Denial of Services (DDoS, see [MR04]) attacks are flooding a specific server with rubbished queries until the system is unable to handle the traffic.

Countermeasures for DDoS includes revocation lists, rate limiting and filtering of ports and service-requests. These techniques can be assigned to privacy aspects as well. Some aspects are already covered by our privacy policy introduced in the previous chapter.

In case of privacy aspects, too many queries can lead to a huge amount of subsets, i.e. $2^n - 1$ subsets for n queries. Even if the server is able to check privacy concerns, the data can be partitioned into small parts such that every subset would leak private information. As a consequence of this, no data is uncovered, although the request is honest.

The policy allows the user to specify purposes and conditions under which a specific application can access private information. This also includes the maximum allowed frequency to query the data set. Thus, designated applications can independently query the data set. The queries for every application are stored in its own sliding window.

5 Conclusion and Further work

Our main objective is to create an environment for privacy preservation in smart environments. In this paper, we introduced an algorithm for monitoring privacy over a period of queries by using a sliding window. Additionally, we considered the setting in which the queries are committed by multiple peers.

Algorithm 1 k-anonymity on multiple queries

```
/**
* Q := a list of queries
* W := sliding window
* n := maximum length of W
* B := a set of combinded queries **/
B = ∅
//Triggered when a new query is deployed to Q
for all new q in Q do
  if |W| < n then
    W.add(q)
  else
    q_old = W.take();
    B.remove(q_old);
  end if
  if |B|! = 0 then
    for all b in B do
      //Combine every known query with the new query or its complement
      B.add(b AND q);
      B.add(b AND NOT(q));
    end for
  end if
  //Combine the new query with the complement of every known query
  B.add(q AND(NOT(b in B)))
  for all b in B do
    if count(b) < k then
      RETURN "NO_PRIVACY";
    end if
  end for
  RETURN anonymized_result(q);
end for
```

Further work needs to be performed to establish our framework. The prototype is currently limited for relational data sets. In Big Data scenarios, the collected data and the associated file formats is often heterogeneous. Research into solving this problem is already under progress. Future work will concentrate on testing the proposed algorithm under real conditions in our Smart Lab.

Acknowledgments.

Hannes Grunert is funded by the German Research Foundation (DFG), Graduate School 1424 (Multimodal Smart Appliance Ensembles for Mobile Applications - MuSAMA). I gratefully acknowledge the constructive comments of the anonymous referees.

References

[BL73] D Elliott Bell and Leonard J LaPadula. Secure computer systems: Mathematical foundations. Technical report, DTIC Document, 1973.

[EK05] José L Encarnaçao and Thomas Kirste. Ambient intelligence: Towards smart appliance ensembles. In *From Integrated Publication and Information Systems to Information and Knowledge Environments*, pages 261–270. Springer, 2005.

[Gru14] Hannes Grunert. Privacy Policy for Smart Environments. `http://www.ls-dbis.de/pp4se`, 2014.

[KG06] Daniel Kifer and Johannes Gehrke. Injecting Utility into Anonymized Data Sets. *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD)*, pages 217–228, 2006.

[LLV07] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *IEEE 23rd International Conference on Data Engineering*, pages 106–115. IEEE, 2007.

[LLZM12] Tiancheng Li Li, Ninghui Li, Jian Zhang, and Ian Molloy. Slicing: A New Approach for Privacy Preserving Data Publishing. *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD)*, 24(3):561–574, March 2012.

[MKGV07] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.

[MR04] Jelena Mirkovic and Peter Reiher. A Taxonomy of DDoS Attack and DDoS Defense Mechanisms. *SIGCOMM Comput. Commun. Rev.*, 34(2):39–53, April 2004.

[SL08] Axel Steinhage and Christl Lauterbach. Sensfloor (r): Ein aal sensorsystem für sicherheit, homecare und komfort. *Ambient Assisted Living-AAL*, 2008.

[Swe02] Latanya Sweeney. k-anonymity. A Model for Protecting Privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, October 2002. doi: 10.1142/S0218488502001648.

[W3C07] W3C. Platform for Privacy Preferences (P3P) Project. `http://www.w3.org/P3P/`, 2007.

[WT13] Benjamin Wagner and Dirk Timmermann. Adaptive clustering for device free user positioning utilizing passive RFID. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pages 499–508. ACM, 2013.