Detecting inference attacks involving sensor data in a multi-database context: Issues & challenges

Paul Lachat, Nadia Bennani, Veronika Rehn-Sonigo, Lionel Brunie, Harald Kosch

Received: 11 February 2022 | Revised: 16 June 2022 | Accepted: 20 June 2022

DOI: 10.1002/itl2.387

LETTER

WILEY

Detecting inference attacks involving sensor data in a multi-database context: Issues & challenges

Paul Lachat[1,2] | **Nadia Bennani**[1] | **Veronika Rehn-Sonigo**[3] | **Lionel Brunie**[1] | **Harald Kosch**[2]

[1]LIRIS, INSA Lyon, Villeurbanne, France

[2]DIMIS, University of Passau, Passau, Germany

[3]FEMTO-ST Institue, University of Bourgogne Franche-Comte, Besançon, France

Correspondence
Paul Lachat, LIRIS, INSA Lyon, Villeurbanne, France.
Email: paul.lachat@insa-lyon.fr

Funding information
Deutsch-Französische Hochschule

Nowadays applications produce and manage data of individual among which some may be sensitive and must be protected. Moreover, with the advent of smart applications, sensor data are produced by IoT devices in a huge quantity and sent to servers in the vicinity to be stored and processed. Meanwhile, newly discovered inference channels involving sensor data gives insights on personal data and raises new threats on individuals privacy. They escape the vigilance of traditional inference detection systems devoted to protecting personal data stored locally in a database. In this paper, we motivate the need of a distributed inference detection system acting in a general multi-database context and we highlight the issues that such a system would face.

KEYWORDS
inference attack detection systems, multi-database, personal data, sensor data

1 | INTRODUCTION

The current ubiquity of personal data implies that collected data are exchanged with data collectors and shared with authorized entities. For the sake of privacy, databases are protected by access control (AC) mechanisms against direct unauthorized access to sensitive personal data. But, due to the inference problem,[1] dishonest users can indirectly access sensitive personal data by exploiting for instance the *dependency strategy*.[2] Therefore, in additional databases can be protected by inference detection systems (InfDSs) against inference attacks.[3-7] However, personal data are rarely stored in a single database but rather spread over multiple databases managed by several distinct entities. In such a multi-database context (MD), a dishonest entity can leverage her authorized accesses to non-sensitive data stored in distinct databases, using the *distributed dependency strategy*[2] to infer sensitive data of individuals. By doing so, the attacker bypasses existing InfDSs proposed in the literature, which each protects a database. A first study of this problem for distinct personal databases has been presented in a previous work.[8] Moreover, with the advent of smart devices, sensor data characterizing physical measurements are issued and collected as data streams. Such data are either directly (eg, by means of wearable sensors[9]) or indirectly (eg, measuring the power consumption of a house[10]) related to an individual. Several works[11] have demonstrated ways to infer personal data using sensor data. Therefore, it becomes possible to infer sensitive personal data by involving non-sensitive personal data and sensor data in a MD context.

To illustrate these privacy threats, let us consider the following motivating example where a recommendation service aims to enable its customers to stay healthy. To do so, the service collects and stores in a database DB_{per} the following information: *Age*, *Sex*, *PAL* (Physical Activity Level) and *CVD* (Cardiovascular Disease) status. In parallel, it collects and stores in a database DB_{sen} the sensor data generated by wearable sensors of customers which it uses to compute their

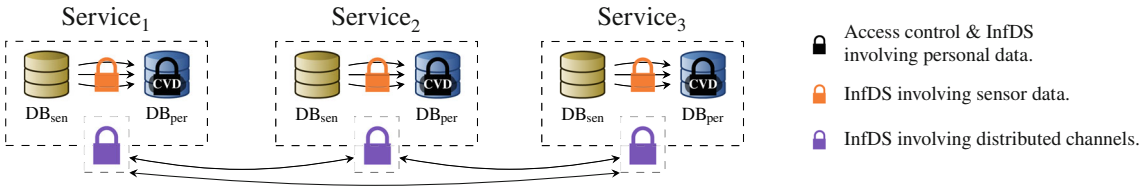

FIGURE 1 Three services have their own local sensitive personal data protected by an InfDS involving personal data and another involving sensor data. The services collaborate via a last InfDS to be protected against distributed inference attacks

activities[9] in order to propose them the most suitable exercises. To protect sensitive personal data in DB_{per}, that is, the *CVD* in our case, the service deploys both an AC mechanism and an InfDS on the DB_{per}. We assume that authorized entities (eg, an insurance employee) can access data from both DB_{per} (except the *CVD*) and DB_{sen}, in order to be able to assess the health risk of each customer and adapt their insurance policies. In such a setting, an authorized but dishonest entity could infer the *CVD* status of a customer *c* with a high enough percentage of confidence, that is, $P_c(\text{CVD}) \geq 50\%$, by leveraging a first background knowledge describing the probability of having a *CVD* based on *Age*, *Sex* and *PAL*[12]. Assuming that the InfDS prevents the exploitation of this inference channel, the attacker queries only the *Age* and *Sex* information from DB_{per}, which allows her to calculate $P_c(\text{CVD}|\text{Age} = 45, \text{Sex} = \text{male}) = 49.4\%$. A second background knowledge,[9] that describes the ability to infer activities performed by a customer based on the data, comes from wearable sensors. She is then able to infer the *PAL* of the targeted customer by querying its sensor data from DB_{sen} which enables her to update her knowledge $P_c(\text{CVD}|\text{Age} = 45, \text{Sex} = \text{male}, \text{PAL} = \text{poor}) = 52.7\%$. Therefore, mining the *PAL* from the sensor data allows the attacker to infer the *CVD* without being detected by the InfDS which only keeps track of queries issued to DB_{per}.

In a more general context, depicted by Figure 1, where multiple services are assumed to own at least a DB_{per} and a DB_{sen}, besides InfDSs devoted to protecting data from inference attacks within personal databases (represented by the black locks on Figure 1), there is a need to both detect at query-time (to preserve the maximum availability to the data[1]):

1. Attacks which leverage both non-sensitive personal data in DB_{per} and knowledge mined from sensor data stored in DB_{sen} (represented by the orange locks on Figure 1).
2. Attacks occurring when a user acquires knowledge querying an external personal or sensor database (represented by the purple locks on Figure 1).

Therefore, in this paper we claim the importance of designing an InfDS able to detect inference attacks (1) in a *multi-database* context (ii) at *query-time* (iii) exploiting dependencies within *personal databases* (iv) and/or data mined from *sensor databases*, and to this purpose we analyze the issues and challenges related to its requirements.

The remainder of this article is organized as follows: Section 2 formulates the issues and their respective challenges that must be dealt with in order to tackle the new inference attacks; Section 3 reviews the related works, describes the limits of each approach with respect to the issues described in Section 2, and compare existing solutions with the requirements (i)-(iv); The conclusion and future research directions are provided in Section 4.

2 | ISSUES & CHALLENGES

In order to detect at query-time inference attacks involving sensor data, one should be able to reason on the existing dependencies that an attacker can leverage, both from databases containing sensor data and/or personal data. Hence, it requires capturing the knowledge gained by the users via queries in a representation which allows such reasoning. Moreover, the MD implies that the detection system must cope with heterogeneous databases owned by distinct services. In the following, we have categorized the issues into three main groups, corresponding to the open issues that must be solved:

- The first one is related to the requirements a model must meet to represent the knowledge a use acquires after querying both personal and sensor data, based on data dependencies.
- The second one presents optimization issues that must be solved to detect inferences at query-time, in an acceptable time, while coping with the huge amount of user knowledge, created by the continuous generation of sensor data.
- The last group describes what an InfDS should achieve to process queries in the MD context, while preserving the privacy of both individuals and services.

2.1 | Modeling issues

Detecting inference attacks involving sensor data leverages some modeling challenges that we detail in this section. To be able to exploit an inference channel implying raw data, the attacker needs to acquire enough knowledge (eg, for example a certain amount of consecutive measures). Consequently, a suitable InfDS should be able to model this new knowledge in a form that allows it to discover the attack. This raises several issues:

A first question is which knowledge should be represented when sensor data are queried; besides data are of stream nature which means very big. It is observed that the data itself (ie, measures) is not so important for the detection but rather the amount of data acquired, the frequency of data and the time this data has been generated by the sensors. Thus, modeling the user knowledge about sensor data requires metadata representation (ie, time window, frequency, and so on).

The second modeling challenge concerns metadata diversity due to the diversity of algorithms behind the inference channels and the data they capture and exploit. This requires a generic way to represent metadata and the capacity of the system to easily integrate new metadata representation, according to the discovery of new inference channels with new metadata requirements.

Lastly, as shown through the motivating example, inference channels based on sensor data can be leveraged to acquire personal data (eg, the PAL). Therefore, the new inference detection system should be able to reason thanks to a unique model integrating both knowledge about obtained personal and sensor data.

2.2 | Efficiency issues

In the next decade, with the advent of applications which manage sensor data of millions of users, services are expected to collect huge amount of data. In addition, the query rate on these data is expected to be very high, which impacts the size of the knowledge managed by an InfDS supporting sensor data. In the meantime, the query time should remain as short as possible. Consequently, the main challenge is how to make the detection time the lowest to keep the query time acceptable.

Behind this challenge, several questions lurk: How would one organize the knowledge storage so that it would be efficient to retrieve? How could the detection algorithms be optimized to target only data required at each step? Is it interesting to consider a distributed inference detection system close to the sensors which generate the data, by exploiting new paradigms (eg, Edge and/or Fog Computing paradigms)? Is it necessary to systematically launch the detection online, keeping in mind that only few queries result in an attack situation? How could we categorize users so that only suspicious users/queries are controlled online?

2.3 | Inference detection attacks issues in the MD context

As explained above, detecting an inference attack in the MD context requires the InfDS to work on a data model that represents data dependencies both inside a database and between attributes of distinct databases.

The first faced issue is how to achieve a semantic matching between attributes of distinct databases. Indeed, the same information could be coded differently in two distinct databases (eg, in one database the first and last name of an individual are stored as a single attribute while in the second database the first name and the last name are represented as two distinct attributes). Furthermore, in the case where the databases are managed by distinct services, the inference detection task must be delegated to an external entity. This generates an additional issue, the external inference detection system needs to have access to (a) the databases schema to be able to build a data model that captures all the data dependencies among attributes, which implies disclosing the database structures and attributes. It also requires that (b) the user's knowledge is managed at the external inference detection system level, that is, the purple locks in Figure 1. Detecting inferences in this case leads to disclosing data at the instance level. For example, we consider a simple situation where DB^1_{per} and DB^2_{per} are owned respectively by services S^1 and S^2 and we assume that DB^1_{per} and DB^2_{per} contain both the same content. Then, if a user u queries *Age* and *Sex* from DB^1_{per}, thanks to the external inference detection system, the *PAL* queried from DB^2_{per} is denied as it would allow the *CVD* to be identified. However, the sensitive information is then disclosed to the detection system itself.

An alternative way to enforce detection without disclosing database schemas and instance information would be to enforce the detection at the side of each service. In fact, the local inference detection systems check the inference condition

locally and collaborate to update the other InfDSs about the knowledge the user gains, in order to avoid answering queries in case of inference attacks. The challenge, in this case, is how to adapt privacy-preserving data exchange protocols (eg, secure Multi-Party protocols) to the type of exchanged data in order to reach the distributed inference detection objective. Besides, the privacy it offers, this solution avoids the huge data exchange due to the database schema exchange and user knowledge at the instance level.

3 | RELATED WORK

In this section, we review the related work in the field of inference detection and raise their limitations to overcome the issues listed above. We can distinguish two categories of systems: those detecting inference attacks at query-time and those detecting and removing inference channels by design. Most of the solutions proposed in the scientific literature aim to protect only a single personal database against inference attacks. Therefore, in the following we present the solutions targeting the protection of a single database then those targeting several databases.

3.1 | Solutions targeting single databases

To detect attacks at query-time, Chen et al.[3] and Guarnieri et al.[5] propose to tackle inference attacks by building models which represent probabilistic inference channels. While Chen et al. propose a mechanism reasoning on the probabilistic dependency among attributes in a single centralized database. Guarnieri et al. propose a system where one module acts as a policy decision point and another checks inference attempts. Moreover, the solution of Guarnieri et al. works under the assumption that only closed queries are issued to the database. All these solutions address inference attack detection considering no updates on the databases which is not suitable to the continuous sensor data generation. Chen et al.'s solution does not offer any possibilities to represent dependencies between sensor data and personal data, with the associated temporal and/or spatial constraints.

The system presented by Toland et al.[4] models the functional dependencies within a database to compute the disclosed knowledge each time a query is issued. In their work, the functional dependencies are limited to logical dependencies. To the best of our knowledge, this work is the only one that considers database updates (ie, tuple update, deletion or insertion), by storing the most recent updates in the query history log. This solution however focuses only logic dependencies in relational databases containing personal data and has not been designed to model sensor data.

To detect attacks by design, Noury et al.[7] propose an access control model which enforces by design value and temporal constraints in time series databases. In their context, an inference occurs when a new access control rule takes precedence on an older one, enabling dishonest users to infer sensitive values. Precedence conflicts are detected thanks to the static rules analysis. However, this model is not adapted to represent dependencies between time series data.

Both Daniels et al.[13] and Qi et al.[14] aim at detecting and preventing inference attacks which occur when personal data are modeled as *resource description framework* (RDF) triplets. Daniels et al. propose a framework called the *pre-release inference analyzer* (PIA) which aims at preventing inference attacks occurring by the simultaneous use of non-sensitive medical data and domain specific ontologies. The PIA annotates the data with the considered ontology, which results in a graph of RDF triplets. With the defined privacy policy, it also annotates triplets considered as sensitive. Before releasing a set of data, it reasons on the set of triplets which are queried to check if a disclosure can occur. If so the PIA can either remove or generalized triplets to prevent users to perform the attack. Qi et al. directly reason on RDF triplets and follows the same reasoning of Daniels et al.'s solution to identify the subset of triplets that must be removed from the answer to prevent unwanted disclosure. Both solutions rely on domain specific ontology to perform the inference reasoning. To the best of our knowledge no ontologies or systems have been proposed to represent or reason on inference channels between sensor and personal data.

3.2 | Solutions targeting several databases

A few solutions are proposed to detect inference attacks in a MC context as described by Jebali et al.[16]

To detect attacks at query-time, Chang et al.[6] propose a solution to model probabilistic dependencies between distributed personal databases in the same system. Similarly to Chen et al.'s solution, Chang et al. work does not consider

TABLE 1 Comparison of the existing solutions w.r.t the requirements (i)–(iv) described in Section 1. The parenthesis describe that a solution satisfies partially a requirement. The following comments provide the reasons of this partial coverage: ❶ The personal data are represented as RDF triplets which make the assumption that a domain specific ontology exist; ❷ The databases are managed by the same system; ❸ Only closed queries on personal data are considered; ❹ The inference is related to the exploitation of access control rules and not dependencies within time series database; ❺ The users do not query databases directly, but via a single entry point

	Multi-database	Query-time	Personal data	Sensor data
Daniels et al.[13]			(✓)❶	
Chang et al.[6]	(✓)❷	✓	✓	
Chen et al.[3]		✓	✓	
Guarnieri et al.[5]		✓	(✓)❸	
Lachat et al.[8]	✓	✓	✓	
Noury et al.[7]				(✓)❹
Qi et al.[14]		✓	(✓)❶	
Sellami et al.[15]	(✓)❺	✓	✓	
Toland et al.[4]		✓	✓	

dependencies between sensor data and personal data. Moreover, they assume that the databases are all controlled by the same system and not by independent entities.

Lachat et al.[8] extend the work of Chen et al. in order to detect inference attacks exploiting multi-database inference channels using data linkage techniques. Their model assumes the databases are static, contain personal data only, and cannot represent dependencies with temporal or contextual constraints. To the best of our knowledge, it is the only work which addresses the inference detection problem for databases managed by distinct entities.

Both at query-time and by design, Sellami et al.[15] consider inference attacks which occur in data integration systems. In such system, a single entry point provides a unique view to the distributed data. Users do not query directly databases. Instead the entry point issues queries and combines the results to obtain the answer. Sellami et al. propose a solution which reason on local privacy policies defined for personal database and a global privacy policy defined for the entry point, in order to derive functional dependencies. The solution follows two phases: at design-time it identifies all the transactions leading to an inference to update privacy policies accordingly; at query-time they keep track of suspicious queries to prevent users to complete a disclosure transaction. Sellami et al. only consider the logical dependencies within personal databases. Hence, their solution is not suitable to represent inference channels stemming from data mining algorithms.

In Table 1, we compare the existing inference detection systems features, in order to detect at query-time inference attacks involving both sensor and personal data in a multi-database context. To the best of our knowledge, we observe that none of the current InfDSs addresses fully the inference detection as described in this paper.

4 | CONCLUSION

In this paper, we have motivated the need to prevent inference attacks in a multi-database context, extending the reasoning to sensor databases. Indeed, the pervasiveness of sensors provides attackers with new sources of data that, if exploited, could breach the privacy of individuals. We have highlighted the issues and challenges which must be tackled in order to propose a system detecting these new inference attacks. As a future work, we first plan to address the modeling of inference channels and user knowledge involving sensor data by using as a case study the MHEALTH dataset.[9] We then plan to propose an InfDS based on this first model, with a first set of optimizations to demonstrate the feasibility of our approach and to evaluate its performance.

As a future work, we first plan to address the modeling of inference channels and user knowledge involving sensor data. We then plan to propose an InfDS based on this first model, to demonstrate the feasibility of our approach and to evaluate its performance. For this purpose, we aim plan to use the MHEALTH dataset[9] as a case study. In the further future, we plan to investigate in two complementary directions: (i) As querying raw data implies the need of a massive amount of stored knowledge and consequently leads to an expensive detection process, we plan to propose (a) data organization strategies to reduce the amount of data needed for the detection and (b) user profiling methods to limit the detection to

suspicious users only (ii) We plan to propose a distributed inference detection system working on several databases owned by distinct services. To do so, a unified model for inference channel representation and data knowledge is mandatory.

FUNDING INFORMATION

This article is part of a thesis supported by the Deutsch-Französische Hochschule.

PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1002/itl2.387.

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

ORCID

Paul Lachat https://orcid.org/0000-0003-1191-4409

REFERENCES

1. Farkas C, Jajodia S. The inference problem: a survey. *ACM SIGKDD Explor Newsl*. 2002;4(2):6-11. doi:10.1145/772862.772864
2. Woodall P, Brereton P. A systematic literature review of inference strategies. *Int J Inf Comput Security*. 2010;4(2):99-117. doi:10.1504/IJICS.2010.034813
3. Chen Y, Chu WW. Protection of database security via collaborative inference detection. *IEEE Trans Knowl Data Eng*. 2008;20(8):1013-1027. doi:10.1109/TKDE.2007.190642
4. Toland TS, Farkas C, Eastman CM. The inference problem: maintaining maximal availability in the presence of database updates. *Comput Secur*. 2010;29(1):88-103. doi:10.1016/j.cose.2009.07.004
5. Guarnieri M, Marinovic S, Basin D. Securing Databases from Probabilistic Inference. *2017 IEEE 30th Computer Security Foundations Symposium (CSF)* 2017: 343–359. doi: 10.1109/CSF.2017.30
6. Chang L, Moskowitz I. A Study of Inference Problems in Distributed Databases. Research Directions in Data and Applications Security: IFIP TC11 / WG11.3 Sixteenth Annual Conference on Data and Applications Security July 28–31, 2002, Cambridge, UK 2003: 191–204. doi: 10.1007/978-0-387-35697-6_15
7. Noury A, Amini M. An access and inference control model for time series databases. *Fut Gener Comput Syst*. 2019;92:93-108. doi:10.1016/j.future.2018.09.057
8. Lachat P, Rehn-Sonigo V, Bennani N. Towards an inference detection system against multi-database attacks. *New Trends Databases Inf Syst*. 2020;1259:199-209. doi:10.1007/978-3-030-54623-6_18
9. Banos O, Villalonga C, Garcia R, et al. Design, implementation and validation of a novel open framework for agile development of Mobile health applications. *Biomed Eng Online*. 2015;14(2) S6:1-20. doi:10.1186/1475-925X-14-S2-S6
10. Eibl G, Engel D. Influence of data granularity on smart meter privacy. *IEEE Trans Smart Grid*. 2015;6(2):930-939. doi:10.1109/TSG.2014.2376613
11. Kröger J. Unexpected inferences from sensor data: a hidden privacy threat in the internet of things. *Internet of Things. Information Processing in an Increasingly Connected WORLD*. 2019;548:147-159. doi:10.1007/978-3-030-15651-0_13
12. Kubota Y, Evenson KR, MacLehose RF, Roetker NS, Joshu CE, Folsom AR. Physical activity and lifetime risk of cardiovascular disease and cancer. *Med Sci Sports Exerc*. 2017;49(8):1599-1605. doi:10.1249/MSS.0000000000001274
13. Daniels M, Farkas C. Health data privacy: a case of undesired inferences. IEEE EMBS International Conference on Biomedical & Health Informatics (BHI) 2018 2018: 291–294. doi: 10.1109/BHI.2018.8333426
14. Qi Y, Zhu T, Ning H. A semantic-based inference control algorithm for RDF stores privacy protection. IEEE International Conference on Intelligence and Safety for Robotics (ISR) 2018 2018: 178–183. doi: 10.1109/IISR.2018.8535628
15. Sellami M, Hacid MS, Gammoudi MM. A FCA framework for inference control in data integration systems. *Distrib Parallel Databases*. 2019;37(4):543-586. doi:10.1007/s10619-018-7241-5
16. Jebali A, Sassi S, Jemai A. Inference control in distributed environment: a comparison study. *Risks Security Internet Syst*. 2020;12026:69-83. doi:10.1007/978-3-030-41568-6_5

How to cite this article: Lachat P, Bennani N, Rehn-Sonigo V, Brunie L, Kosch H. Detecting inference attacks involving sensor data in a multi-database context: Issues & challenges. *Internet Technology Letters*. 2022;e387. doi: 10.1002/itl2.387