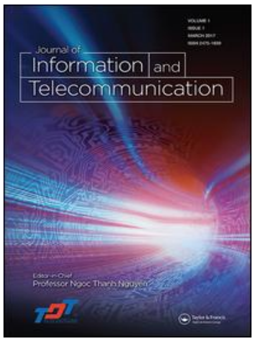

Taylor & Francis
Taylor & Francis Group

Journal of Information and Telecommunication

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tjit20

Secure data outsourcing in presence of the inference problem: issues and directions

Adel Jebali , Salma Sassi & Abderrazak Jemai

To cite this article: Adel Jebali , Salma Sassi & Abderrazak Jemai (2020): Secure data outsourcing in presence of the inference problem: issues and directions, Journal of Information and Telecommunication, DOI: 10.1080/24751839.2020.1819633

To link to this article: https://doi.org/10.1080/24751839.2020.1819633

Published online: 24 Sep 2020.

JOURNAL OF INFORMATION AND TELECOMMUNICATION
https://doi.org/10.1080/24751839.2020.1819633

ĐẠI HỌC TÔN ĐỨC THẮNG
TON DUC THANG UNIVERSITY

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS

Check for updates

Secure data outsourcing in presence of the inference problem: issues and directions

Adel Jebali[a], Salma Sassi[b] and Abderrazak Jemai[c]

[a]Faculty of Mathematical Physical and Natural Sciences of Tunis, SERCOM Laboratory, Tunis El Manar University, Tunis, Tunisia; [b]Faculty of Law Economics and Management of Jendouba, VPNC Laboratory, Jendouba University, Jendouba, Tunisia; [c]Polytechnic School of Tunisia, SERCOM Laboratory, Carthage University, INSAT, Tunis, Tunisia

ABSTRACT

With the emergence of cloud computing paradigms, secure data outsourcing has become one of the crucial challenges which strongly imposes itself. Data owners place their data among cloud service providers in order to increase flexibility, optimize storage, enhance data manipulation and decrease processing time. Nevertheless, from a security point of view, access control is a major challenge in this situation seeing that the security policy of the data owner must be preserved when data is moved to the cloud. Nonetheless, the lack of a comprehensive and systematic review motivated us to construct this reviewing paper on this research problem. Here, we discuss current and emerging research on privacy and confidentiality concerns in data outsourcing and pinpoint potential issues that are still unresolved.

ARTICLE HISTORY
Received 13 January 2020
Accepted 2 September 2020

KEYWORDS
Cloud computing; data outsourcing; access control; inference leakage; secrecy and privacy

1. Introduction

In light of the growth volume and variety of data from diverse sources, including health systems, social insurance systems, scientific and academic data systems, smart cities and social networks, in-house storage and processing of large collections of data has becoming very cost. Hence, modern database systems have evolved from a centralized to a distributed storage architecture, which have given emergence to the *Database- as-a-Service* paradigm. Data owners place their data among cloud service providers (CSP) in order to increase flexibility, optimize storage, enhance data manipulation and decrease processing time. Nonetheless, security concerns are widely recognized as a major barrier to cloud computing and other data outsourcing or Database-as-a- Service arrangements. Users are reluctant to place their sensitive data in the cloud due to concerns about data disclosure to potentially untrusted external parties and other malicious parts (Xu et al., 2015). Being processed and stored externally, owners cannot anymore take control of their sensitive data, consequently users privacy will be at risk. From this secure perspective, access control is a major challenge seeing that the security policy of data owner must be

CONTACT Adel Jebali adel.jbali@fst.utm.tn Faculty of Mathematical Physical and Natural Sciences of Tunis, SERCOM Laboratory, Tunis El Manar University, Tunis 1068, Tunisia

preserved when data is moved to the cloud. Access control policies are enforced in cloud service providers level by keeping some sensitive data separated from each other (Samarati & Di Vimercati, 2010). Besides, some other technique could be helpful to guarantee the confidentiality of data like encryption (Biskup & Preuß, 2013; Bkakria et al., 2013b; Ciriani et al., 2007). This latter is used to break sensitive associations among outsourced data by encrypting some attributes. Additionally, security breaches in distributed cloud databases could be exacerbated due to inference leakage. This latter is produced when malicious user combines the legitimate response that he received from the system with metadata. During the last two decades, researchers have devoted a lot of efforts to enforce access control policies and privacy protection requirements externally while maintaining balance with data utility 2017 (Aggarwal et al., 2005; Alsirhani et al.,; Bollwein & Wiese, 2017, 2018; Ciriani et al., 2009a, 2009b; di Vimercati et al., 2014).

We give in this paper a review study of current and emerging research on privacy and confidentiality concerns in data outsourcing and highlighting research directions in this field. In summary, our systematic review treats security concerns in cloud database systems for both communicating and non-communicating servers. Besides, it surveys this research field in relation with the inference problem. As a consequence, many unresolved problems were introduced. Parting from these limits, we propose an overview of our proposed solution (because it is an ongoing work) to firstly optimize data distribution without the need to the query workload. Secondly the partition of the database in the cloud by taking into consideration access control policies and data utility, and finally we propose a query evaluation model on a big data framework to securely processing distributed queries while retaining access control. The reminder of this paper is organized as follows: section 2 describes the reviewing methodology adopted in this paper. Section 3 reviews emerging research on data outsourcing in presence of privacy concerns and data utility. Section 4 discusses data outsourcing in relation with the inference problem. In section 5 we introduce our proposed solution to implement a secure distributed cloud database on a big data framework (Apache Spark). Research directions are given in section 6. Finally, we conclude in section 7.

2. The reviewing methodology

The methodology of reviewing adopted in this paper follows the checklist proposed by the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) (Moher

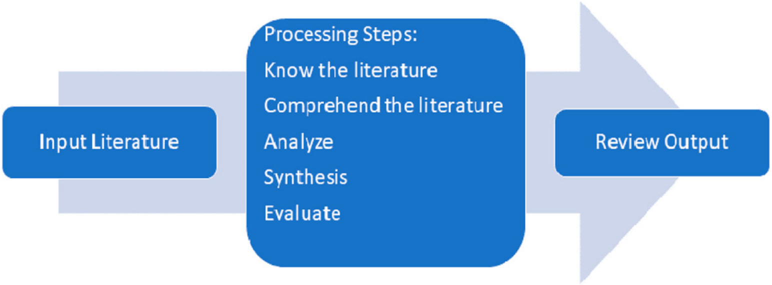

Figure 1. Three stages of literature review process.

Table 1. Keywords used in review search.

Keyword	Number of viewed papers
Access control, Data outsourcing	43
Cloud computing, Authorization policies	78
Database, inference leakage	33
Confidentiality constraints, Cloud database	41
Secure data integration	11
Big data, Distributed query processing	39
Privacy, data publishing	24

et al., 2009). It includes as shown in Figure 1 three steps: *Input literature, processing steps and review output*.

2.1. Input literature

In this section we describe selected literature and their selection process. Firstly, our advance keyword research was conducted on Google Scholar search engine with time filter from 1 January 1990 to 31 December 2019. Table 1 listed keywords used in different queries search in Google Scholar.

The logical operator used between keywords during search was the 'And' operator. Finally, from the 269 viewed papers, 43 articles were retained for review. We give their distribution by year in Figure 2.

2.2. Processing steps

All along the review, papers were processed by firstly identifying the problem, then understanding the proposed solution process and finally listing the important findings. We summarized and compared each paper with the papers associated with the similar problem. Besides, for each processed paper a critical of three or four sentences was introduced to highlight the limits and specify potential directions that may be followed to enhance the proposed approaches. Hence, based on our literature review we have classified papers into 3 categories as shown in Figure 3. The first category of papers treats secure

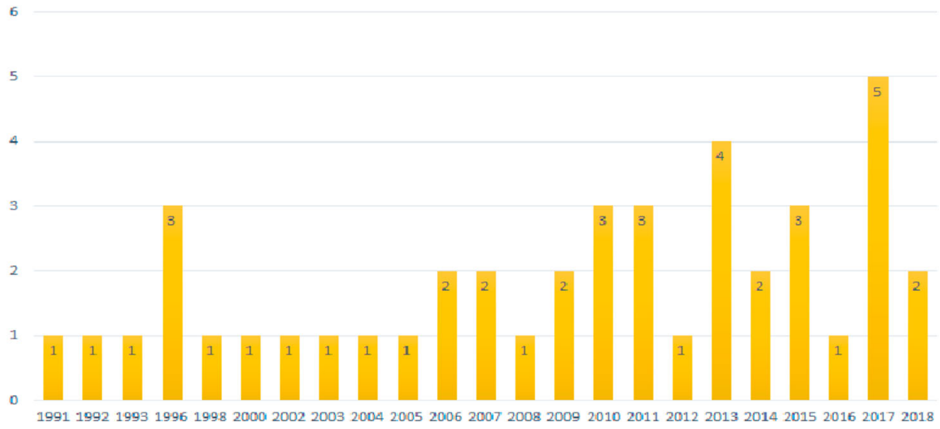

Figure 2. Distribution of publication by year (43 articles).

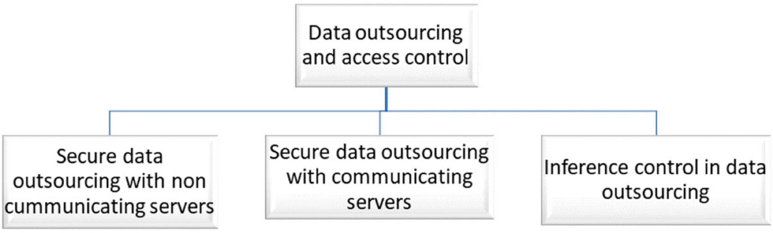
Data outsourcing and access control
Secure data outsourcing with non cummunicating servers
Secure data outsourcing with communicating servers
Inference control in data outsourcing

Figure 3. Classification scheme of the literature.

data outsourcing problem when the servers in the cloud are unaware of each other. The second category takes into consideration the interaction between servers and how this later can aggravate the situation. In the last category, we tackled data outsourcing in relation with inference problem since this later can exploit semantic constraints to bypass authorization policies in the cloud level.

2.3. Review output

The outcome of the methodological review process was presented in section 5 and 6. In section 5 we have presented an incremental approach composed of 3 steps, each step treats one of the three problems mentioned in the previous section. We believe that our proposed solution can give good results compared to other reviewed approaches. Moreover, in section 6 we report other potential future research areas.

3. Preserving confidentiality in data outsourcing scenarios

Among security researchers community, there is a consensus about the efficiency of data outsourcing for solving data management problems (Samarati & Di Vimercati, 2010). This later consists of moving data from in-house storage to cloud databases while maintaining a balance between data confidentiality and utility (Figure. 4).

Cloud service providers are considered *honest-but-curious:* the database servers answer user-queries correctly and do not manage stored data, but they attempt to analyze intelligently data and queries in order to learn as much information as possible from them.

Two powerful techniques have been proposed to enforce access control in cloud databases: the first one is by exploiting vertical database fragmentation to keep some sensitive data separated from each other. The second one is by resorting to encryption to make single attribute invisible to unauthorized users. These two techniques can be implemented using the following approaches:

- Full outsourcing (Aggarwal et al., 2005): The hole in-house database is moved to the cloud. It considers vertical database fragmentation to enforce confidentiality constraints with more than two attributes by keeping them separated from each other among distributed servers. Moreover, it resorts to encryption in order to hide confidentiality constraints with single attribute.
- Keep a few (Ciriani et al., 2009b): this approach departs from encryption by involving owner side. The attributes to be encrypted are stored in plain text in the owner side

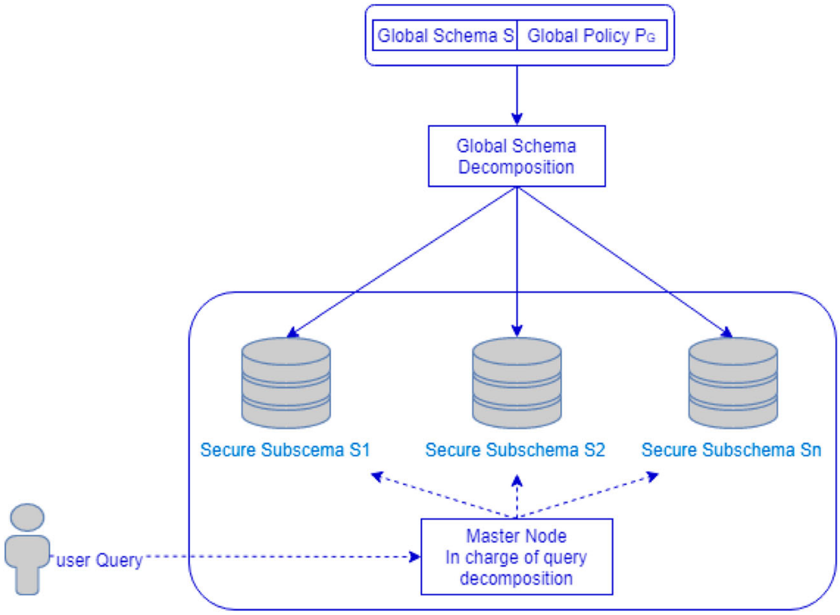

Figure 4. Secure data outsourcing.

since this later is considered as a trusted part. the rest of the database is distributed among servers while maintaining data confidentiality through vertical fragmentation.

Parting from the fact that Encrypting data for storing them externally carried a considerable cost (Samarati & Di Vimercati, 2010), previous studies have primarily concentrated on non-communicating cloud servers (Aggarwal et al., 2005; Ciriani et al., 2009a, 2009b; De Capitani di Vimercati et al., 2010; di Vimercati et al., 2014). In this situation servers are unaware of each other and do not exchange any information. When a master node receives a query it decomposes it and processes it locally without the need to perform join query. In recent years, researchers have studied the effect of communication between servers on query execution and secure query evaluation strategies have been elaborated (Bkakria et al., 2013a, 2013b; De Capitani di Vimercati et al., 2016; di Vimercati et al., 2013). In the rest of this section we will discuss current and emerging research efforts in each of the mentioned architectures.

3.1. Secure data outsourcing with non-communicating servers

The first work attempting to enforce access control in database outsourcing using vertical fragmentation was presented in (Aggarwal et al., 2005). Under the assumption that servers do not communicate, the work aims to split the database on two untrusted servers while preserving data privacy with some of the attributes possibly encrypted. Hence, a secure fragmentation of a relation R is a triple (F_1, F_2, E) where F_1, F_2 contain attributes in plain text stored in different servers and E is the set of encrypted attributes. The tuple identifier and the encrypted attributes are replicated with each fragment. The protection measures was also augmented by a query evaluation technique defining how queries on the original table can be transformed into queries on the fragmented table. The work in Hudic et al. (2012) introduces an approach to enforce confidentiality and privacy while outsourcing data to CSP. The proposed technique relies on vertical fragmentation and applies only minimal encryption to preserve data exposure to malicious part. However, the fragmentation algorithm enforces the database logic schema to be in third normal form to produce a good fragmentation design, also the query execution cost was not proven to be minimal. Authors of Ciriani et al. (2007) address the problem of privacy preserving data outsourcing by resorting to the combination of fragmentation and encryption. The former is exploited to break sensitive associations between attributes while the latter enforces privacy of singleton confidentiality constraints. Besides, authors define a formal model of minimal fragmentation and propose a heuristic minimal fragmentation algorithm to efficiently execute queries over fragments while preserving security properties. Meanwhile, when a query executed over a fragment involving an attribute that is encrypted, further query will be executed to evaluate conditions of the attributes and this will lead to a performance degradation by slowing down query processing.

In Ciriani et al. (2011) researchers treat the concept of secure data publishing in presence of confidentiality and visibility constraints. By modelling these two latters as Boolean formulas and fragment as complete truth assignment, authors rely on Ordered Binary Decision Diagrams (OBDD) technique to check whether a fragmentation satisfies confidentiality and visibility constraints. The proposed algorithm runs throw OBDD and returns a fragmentation that guarantees correctness and minimality. Nonetheless, query execution cost was not investigated in this paper and the algorithm runs only on database schema with single relation. Authors of Xu et al. (2015) studied the problem of finding secure fragmentation with minimum cost for query support. Firstly, they define the cost of a fragmentation F as the sum of the cost of each query Qi executed on F multiplied by the execution frequency of Qi. Secondly, they resort to heuristic local search graph-based approach to obtain near optimal fragmentation. The search space was modelled as a fragmentation graph and transformation between fragmentation as a set of edges E. Then, two search strategies where proposed: a static search strategy which is invariant with the number of steps in a solution path, in addition to a dynamic search strategy based on guided local search which guarantees the safeness of the final solution while avoiding dead-end. However, this paper does not investigate visibility constraints which is an important concept for data utility. Moreover, other heuristic search techniques could be addressed (Tabu search or simulated annealing).

The work in Ciriani et al. (2009b) puts forward a new paradigm to securely publishing data in the cloud while completely departing from encryption since this latter is sometimes considered a very rigid tool, delicate in its configuration, and may slowing down query processing. The idea behind this work is to engage owner side (assumed to be a trusted part) to store a limited portion of data (supposed to be encrypted) in the clear and use vertical fragmentation to break sensitive associations among data to be stored in the cloud. The proposed algorithm computes a fragmentation solution that minimizes the load for the data owner while guaranteeing privacy concerns. Moreover, authors highlight other metrics that can be used to characterize the quality of a fragmentation and decide which attribute is affected to client side and which attribute is externalized. Even though, engaging the client to enforce access control requires the fact to mediate every query in the system which could lead to bottleneck and impacting performances.

Researchers in Bollwein and Wiese (2017) propose a separation of duties technique based on vertical fragmentation to address the problem of confidentiality preserving when outsourcing data to CSP. To capture privacy requirement confidentiality constraints and data dependencies where introduced in this work. The separation of duties problem is treated as an optimization problem to maximize the utility of the fragmented database and to enhance the query execution over the distributed servers. However, the optimization problem was addressed only from the point of minimizing the number of distributed servers. Besides, when collaboration between servers is established the separation of duties approach is no longer efficient to preserve confidentiality constraints. The NP- Hardness proofness of the separation of duties problem discussed in Bollwein and Wiese (2017) was proven in Bollwein and Wiese (2018). The separation of duties problem was addressed as an optimization problem by the combination of the two famous NP-Hard problems: Bin packing and Vertex coloring. The bin packing problem was introduced to take into consideration capacity constraint of the servers in view that fragments should be placed in a minimum number of servers without exceeding the maximum capacity. Meanwhile, vertex coloring was introduced to enforce confidentiality constraints seeing that the association of certain attributes in the same server violates confidentiality propriety. We would like to mention that this paper studies the separation of duties problem for single relation database and to make the theory applicable in practical scenarios many relations database should be treated.

Parting from the fact that communication between distributed servers in data outsourcing scenarios exacerbates privacy concerns, secure query evaluation strategies should be adopted. In the next subsection we investigate works turning around secure data outsourcing with communicating servers.

3.2. Secure data outsourcing: the case of communicating servers

In the last years, few works had investigated the problem of data outsourcing with communicating servers (Bkakria et al., 2013a, 2013b; De Capitani di Vimercati et al., 2016; di Vimercati et al., 2013). Besides to guarantee confidentiality and privacy preserving when distributing databases in the cloud, these works implemented secure query evaluation strategies to retain access control policy when servers communicate with each other. It is clear when servers (containing sensitive attributes whose association is forbidden) interact through join queries, user's privacy will be at risk. Therefore, secure query evaluation strategies aim to prevent linkability between sensitive attributes when a malicious user tends to establish it.

Authors in Bkakria et al. (2013a) have built on Bkakria et al. (2013b) to propose an approach that securely outsourcing data based on fragmentation and encryption. It also enforces access control when querying data by resorting to query privacy technique. The approach treated the case of multi-relations database and a new inter-table confidentiality constraints was introduced. It assumes that the distributed servers could collude to break data confidentiality so the connection between servers is supposed to be based on primary-key/foreign key. In addition, the query evaluation model which is based on private information retrieval ensure data unlinkability performed by malicious user using semi join query. Even though, the proposed technique enforces database schema to be normalized, and generates a huge number of confidentiality constraints due to the transformation of inter-table constraints to singleton and association constraints which could affect the quality of the fragmentation algorithm. In addition, more generic queries should be considered.

Join query integrity check had been tackled through the work in De Capitani di Vimercati et al. (2016) where researchers inspired from di Vimercati et al. (2013) to propose a new technique in order to verify the integrity of join queries computed by potentially untrusted cloud providers. Authors aim also in their approach to prevent servers to learn from answered queries which could lead to breach user's privacy. To do so, authors show first how markers, twins, salts and buckets can be adapted to preserve integrity when a join query is executed as a semi-join, then they introduce two strategies to minimize the size of the verification: limit the adoption of buckets and salts to twins and markers only and representing twins and markers through slim tuples. Besides, authors demonstrate through their experiments how the computational and communication overhead can be limited due to integrity check.

3.3. Discussion

To summarize, we can classify discussed approaches according to the following criterion: confidentiality constraints support, optimal distribution support and secure query evaluation strategy support. We would like to mention that optimal distribution is treated through secure distributions that guarantee minimum query execution costs over fragments. From this point, it is clear that all mentioned approaches support access control verification through confidentiality constraints. However, query evaluation have not been tackled in all works (Bollwein & Wiese, 2017; Ciriani et al., 2007, 2009b, De Capitani di Vimercati et al., 2010). Those approaches differ from the fact that some of them ensure minimum query execution costs and data utility for the database application, but other ones addressed the problem of data outsourcing with confidentiality constraints only. Besides, among the secure database distribution with query evaluation strategy, we see that the work in Bkakria et al. (2013b) provides an integral framework ensuring secure database fragmentation and communication between distributed servers. Also, it shows a reasonable query execution cost.

Nevertheless, the work assumes that the threat comes from the cloud service providers that try to collude to break sensitive association between attributes, it does not treat the case of internal threat where a malicious user aims to bypass access control with an inference channel. This is why we present in the next section an insightful discussion about data outsourcing in presence of the inference problem.

4. Data outsourcing and the inference problem

Access control models protect sensitive data from direct disclosure via direct accesses, however they fail to prevent indirect accesses (Farkas & Jajodia, 2002). Indirect accesses via inference channels occur when a malicious user combines the legitimate response that he received from the system with metadata, Figure 5. According to Guarnieri et al. (2017), external information to be combined with data in order to produce an inference channel could be database schema, system's semantics, statistical information, exceptions, error messages, user defined functions and data dependencies.

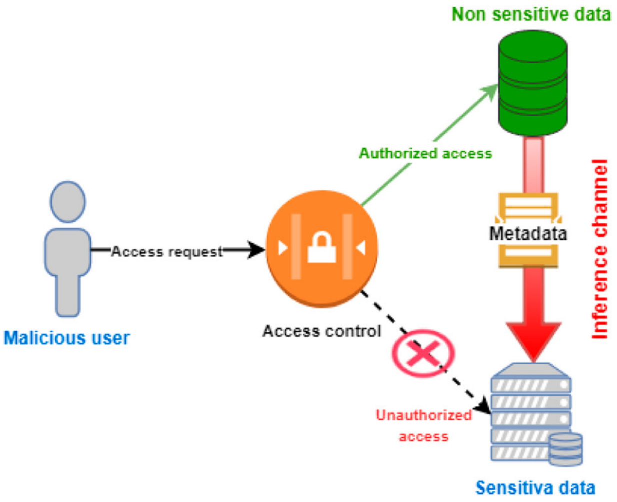

Figure 5. Bypass access control with inference channel.

Table 2. Access control vs inference control.

Access control	Inference control
Direct access control	Indirect access control
Deterministic	Related to stochastic channels
Static: through a set of rules	Dynamic: vary through time and influenced by user action and queries
Normal expensive	More expensive then access control
Computational efficiency and high accuracy of security control	Efficiency and accuracy less than access control
Modular: can cover distributed environment	Adaptability to data distribution requires complicated techniques

Although access control and inference control share the same goal of preventing data from unauthorized disclosure, they differ in several fundamental aspects (Katos et al., 2011). We give in Table 2 the major differences between them. According to our comparison Table 2, we can note that access control is more preferable than inference control from a complexity perspective. Consequently, several researchers have attempted to replace inference control engines with access control mechanisms. We refer the interested reader to Biskup et al. (2008), Biskup et al. (2010), Katos et al. (2011), the discussion of these approaches is out of the scope of this paper.

4.1. Inference attacks and prevention methods

According to Farkas and Jajodia (2002), there are three types of inference attack: Statistical attacks, semantic attacks and inference due to data mining. For each of the mentioned techniques, researchers have devoted a lot of efforts to deal with inference problem. For statistical attacks, techniques like Anonymization and Data-perturbation have been developed to protect data from indirect access. For security threats based on data mining, techniques like privacy-preserving data mining and Privacy-preserving data publishing was carried out. Furthermore, a lot of works have investigated the semantic attacks (Brodsky et al., 2000; Chen & Chu, 2006; Su & Ozsoyoglu, 1991).

There exist in the literature more than one criterion to classify approaches that deal with inference. One proposed criterion is to classify these approaches according to data level and schema level (Yip & Levitt, 1998). In such classification, inference constraints are classified into schema constraints level and data constraints level. Another criterion could be according to the time when the inference control techniques are performed. According to this criterion, the proposed approaches are classified in three categories: design time (Delugach & Hinke, 1996; Hinke & Delugach, 1992; Rath et al., 1996; Wang et al., 2017) and query run time (An et al., 2006; Brodsky et al., 2000; Chen & Chu, 2006; Thuraisingham et al., 1993). The purpose of inference control at design time is to detect inference channels from earliest stage and eliminate them. These approaches provide a better performance for the system since no monitoring module is needed when the users query the database, by consequence improving query execution time. Nevertheless, design time approaches are too restrictive and may lead to over classification of the data. Besides, it requires that the designer has a good concept of how the system will be utilized. On the other hand, run time approaches provide data availability since they monitor the suspicious queries at run time. However, run time approaches lead to performance degradation of the database server since every query needs to be checked by the inference engine. Furthermore, the inference engine needs to manage a huge number of log files and users. As a result, this could induce slowing down query processing. In addition, run time approaches could induce a non-deterministic access control behaviour (users with the same privileges may not get the same response).

From this perspective, we can conclude that the main evaluation criterion of these techniques is a trade-off between availability and system performance. Some works have been elaborated to overcome these problems especially for run time approaches. Example in Yang et al. (2007) a new paradigm of inference control with trusted computing was developed to push the inference control from server side to client side in order to mitigate the bottleneck on the database server. Furthermore, in Staddon (2003) the authors have developed a run time inference control techniques while retaining fast query processing. The idea behind this work was to make query processing time depends on the length of the inference channel instead of user query history.

4.2. Inference control in cloud data integration systems

Data outsourcing and the inference problem is a research field that researchers have begun to investigate few years ago (Biskup et al., 2011; de Mantaras & Saina, 2004; di Vimercati et al., 2014; Haddad et al., 2014; Sayah et al., 2015; Sellami et al., 2015; Turan et al., 2017, 2018). Inference leakage is recognized as a major barrier to cloud computing and other data outsourcing or Database-As-a-Service arrangements. The problem is that the designer of the system cannot anticipate the inference channels that arise in cloud level and could lead to security breaches. Authors in de Mantaras and Saina (2004) pinpoint the inference that occurs in homogeneous peer agent through distributed data mining and call this process peer-to-peer agent based data mining systems. They assert that performing Distributed Data Mining (DDM) in such extremely open distributed systems exacerbates data privacy and security issues. As a matter of fact, inference occurs in DDM when one or more peer sites learn any confidential information (model, patterns, or data themselves) about the dataset owned by other peers during a data mining session. The authors firstly classified inference attacks in DDM in two categories:

- **Inside Attack Scenario:** it occurs when a peer tries to infer sensitive information from other peers in the same mining group. Depending on the number of attackers the authors distinguish single attack (when one peer behaves maliciously) and coalition attack (when many sites collude ta attack one site). Moreover, a probe attack was introduced by the authors, which is independent of the number of peers participating in the attack.
- **Outside Attack Scenario:** it takes place when a set of malicious peer try to infer useful information from other peers in a different mining group. In this case eavesdropping channel attack is performed by malicious peers to steal information from other peers.

After identifying DDM inference attacks, the authors propose an algorithm to control potential attacks (inside and outside attacks) to particular schema for homogeneous distributed clustering, known as *KDEC*. The main idea behind this algorithm is to reconstruct the data from the kernel density estimates since a malicious peer can use the reconstruction algorithm to infer non-local data. However, the algorithm proposed by the authors need to be improved from an accuracy point to expose further possible weakness of the KDEC schema.

Inference control in cloud integration systems has been investigated in the last decade through the work of Haddad et al. (2014), Sayah et al. (2015), Sellami et al. (2015). In such systems, a mediator is defined as a unique entry point to the distributed data sources. It provides to the user a unique view of the distributed data. From a security point of view, access control is a major challenge in this situation since the global policy of the mediator in the cloud level must comply with the back-end data sources policies in addition to possibly enforcing additional security properties Figure 6. The problem is that the system (or the designer of the system) cannot anticipate the inference channels that arise due to the dependencies that appear at the mediator level.

The first work attempting to control inference in data integration systems was introduced in Haddad et al. (2014). The authors propose an incremental approach to prevent inference with functional dependencies. The proposed methodology includes three steps:

- **Synthesizing global policies:** derives the authorization rule of each virtual relation individually by the way that it preserves the local authorization of the local relations involved in the virtual relation.
- **Detection phase:** by resorting to a graph-based approach, this step aims at identifying all the violations that could occur using functional dependencies. Such violation is called violating transaction consisting of a series of innocuous queries when it is achieved leads to violation of authorization rule.
- **Reconfiguration phase:** in this phase the author proposes two methods to forbid the completion of each transaction violation. The first one uses a historic based access control by keeping track of previous queries to evaluate the current query (this method is considered as a run-time approach). The second one proposes to reconfigure

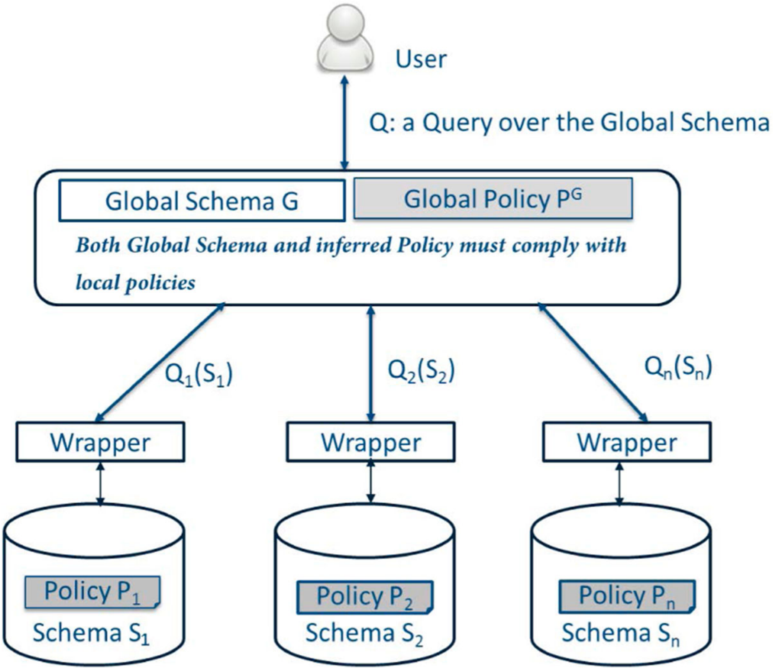

Figure 6. Secure data integration system.

the global authorization policies at the mediator level in a way that no authorization violation will occur (this method is considered at design-time of the global security policy).

In this work, the authors have discussed only semantic constraints due to functional dependencies. Neither inclusion nor multivalued dependencies was investigated. Besides, other mapping approaches need to be discussed such as LAV and GLAV approaches.

The authors of Sellami et al. (2015) have been inspired by Haddad et al. (2014) to propose an approach aiming to control inference in cloud integration systems. The proposed methodology resorts to formal concept analysis as a formal framework to reason about authorization rules and functional dependencies as a source of inference. Authors adopt an access control model with authorization views and propose an incremental approach with three steps:

- **Generation of the global policy, global schema and global FD:** this step takes as input a set of source schema together with their access control policies and starts by translating the schema and policies to formal contexts. Then, the global policy is generated in a way that the source rules are preserved at the global level. Next the schema of the mediator (virtual relations) is generated from the global policy to avoid useless attributes combination (every attribute in the mediator schema is controlled by the global policy). Finally, a global FD is considered from the source FD as a formal context.

- **Identifying disclosure transactions:** by resorting to FCA as a framework to reason about the global policy, the authors identify the profiles to be denied from accessing sensitive attributes at the mediator level. Then, they extract the violating transaction by reasoning about the Global FD.
- **Reconfiguration phase:** this step is achieved by two ways. At design time with a policy healing consisting to complete the global policy with additional rules in order that no violating transaction is achieved. At query run time with a monitoring engine to prohibit suspicious queries.

Authors in Sayah et al. (2015) have examined inference that arises in the web through RDF store. They propose a fine-grained framework for RDF data, then they exploit close graph to verify the consistency propriety of an access control policy when inference rules and authorization rules interact. Without accessing the data (at policy design-time), the authors propose an algorithm to verify if an information leakage will arise given a policy P and a set of inference rules R. Furthermore, the authors demonstrate the applicability of the access control model using a conflict resolution strategy (most specific takes precedence).

4.3. Inference control in distributed cloud database systems

In Biskup et al. (2011) authors resort to a Controlled Query Evaluation strategy (CQE) to detect inference based on the knowledge of non-confidential information contained in the outsourced fragments and prior knowledge that a malicious user might have. Regarding that CQE relies on logic-oriented view on database systems, the main idea of this approach is to model fragmentation logic-oriented too allowing for inference proofness to be proved formally even the semantic database constraints that an attacker may hold. Besides, vertical database fragmentation technique was considered by authors in di Vimercati et al. (2014) to ensure data confidentiality in presence of data dependencies among attributes. Those dependencies allow unauthorized users to deduce information about sensitive attributes. In this work, three types of confidentiality violations that can be caused by data dependencies were defined: firstly when a sensitive attribute or association is exposed by the attributes in a fragment. Secondly, if an attribute appearing in a fragment is also derivable from some attributes in another fragment, thus enabling linkability among such fragments, and thirdly when an attribute is derivable (independently) from attributes appearing in different fragments, thus enabling linkability among these fragments. To tackle these issues, authors reformulate the problem graphically through a hyper-graph representation and then compute the closure of a fragmentation by deducing all information derivable from its fragments via dependencies to identify indirect access. Nevertheless, the major limit of this approach is that it explores the problem only in single relational database.

Despite data outsourcing was not explicitly mentioned in Turan et al. (2018, 2017), these two works aim to control inference problem caused by functional dependencies and meaningful join proactively by decomposing the relational logical schema into a set of views (to be queried by the user) where inference channels cannot appear. In Turan et al. (2017) authors propose a proactive and decomposition-based inference control strategy for relational databases to prevent access to forbidden set via direct or indirect access. The proposed decomposition algorithm controls both functional and probabilistic dependencies by breaking down those leading to infer a forbidden attribute set. However, this approach was considered to rigid by the fact that if the ways of associate forbidden sets attributes are define as a chain of functional dependencies, the algorithm breaks these chains from both sides for both attributes. Parting from this limit, the same researchers propose a graph-based approach in Turan et al. (2018) consisting of proactive decomposition of the external schema, in order to satisfy both the forbidden and required associations of attributes. In this work, functional dependencies are represented as a graph in which vertices are attributes and edges are functional dependencies. Inference channel is then defined as a process of searching a sub-tree in the graph containing the attributes that need to be related. Compared to the approach (Turan et al., 2017), in this one the cut of the inference channel is relaxed by cutting the chains only at a single point, consequently minimizing dependencies loss. Nevertheless, like the previous technique it leads to semantic loss and need query rewriting techniques to query decomposed views.

5. Proposed solution

We present in this section an approach that relies on the relational model and it aims as shown in Figure 7 at producing a set of secure sub-schemas, each sub-schema is represented by a partition P_i and each partition is stored exactly in one server in the CSP. In addition, it introduces a secure distributed query evaluation strategy to efficiently request data from distributed partitions while retaining access control policies. To do that, our methodology takes as input a set of functional dependencies (FD), a relational schema R and applies the following phases:

(1) ***Constraints generation:*** This phase aims at generating two types of constraints that in addition to the confidentiality constraints will guide the process of vertical schema partitioning. This will be done through the following steps:
   - *Visibility constraints generation*: these constraints will be enforced as *soft constraints* in the partitioning phase their severity is less than confidentiality constraints. To generate them, we perform a semantic analysis of the relational schema in order to detect semantic relatedness between attributes and users roles. These constraints should be preserved (stay visible) when the relational schema is fragmented (in other words we aim to maximize intra-dependency between attributes that seem to be frequently accessed by the same role while minimizing the inter-dependency between attributes in separate parts).
   - *Constraint-based inference control generation*: these constraints are enforced (like confidentiality constraints) as *hard constraints*. In this step we resort to the method proposed in Turan et al. (2018) to build functional dependencies graph and generate join chains. Then, we use a relaxed technique to cut the join chains only at a single point in order to minimize dependencies loss. We mean by cutting a join chain at a single point, the enforcement of the attributes in the *LHS* and *RHS* of the functional dependency representing the cut point as a confidentiality constraint. By consequence, we guarantee that the join chain is broken.

(2) Schema partitioning: In this phase, we resort to hypergraph theory to represent the partitioning problem as a hypergraph constraint satisfaction problem. Then, we

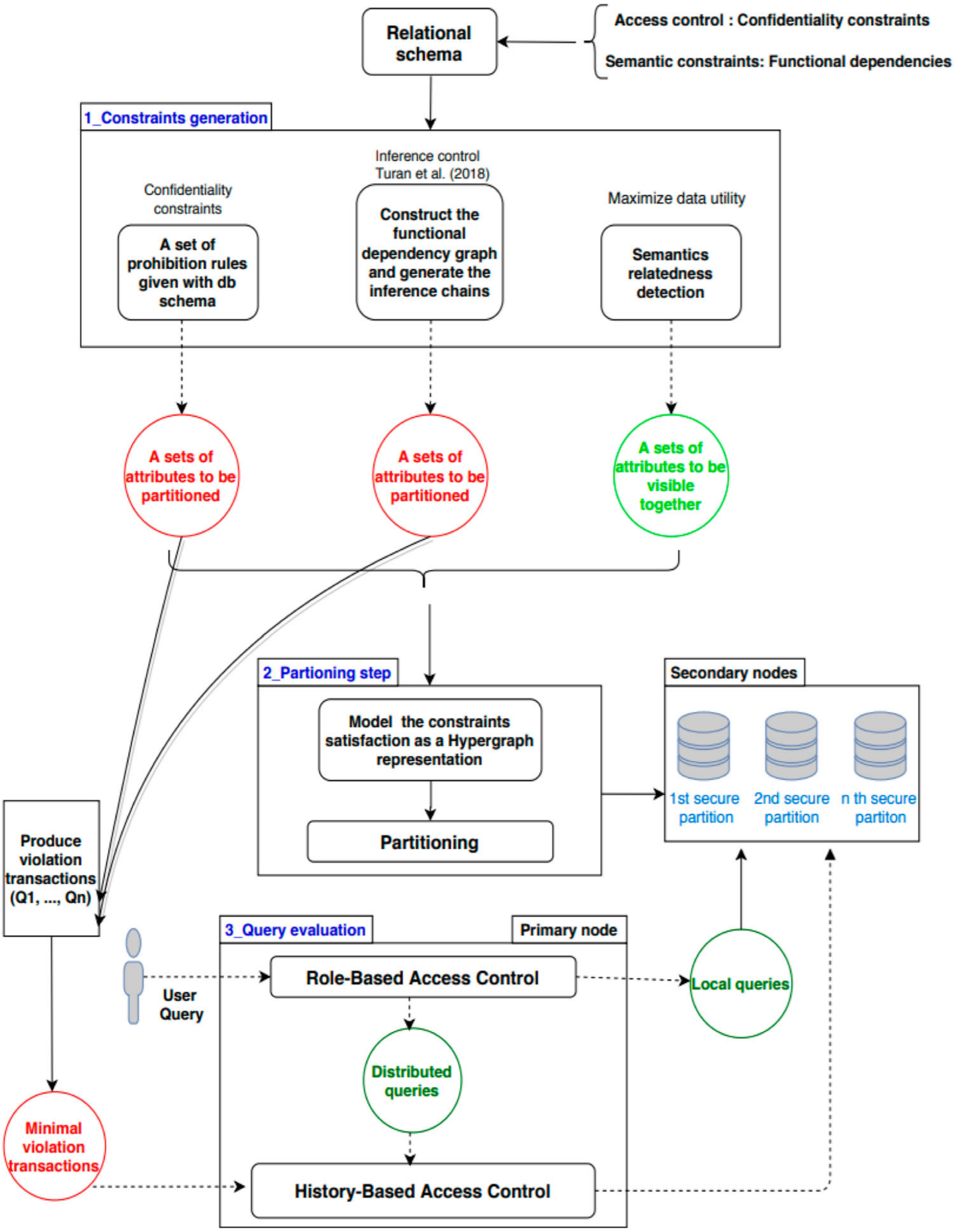
Relational schema
Access control : Confidentiality constraints
Semantic constraints: Functional dependencies
1_Constraints generation
Confidentiality constraints
A set of prohibition rules given with db schema
Inference control Turan et al. (2018)
Construct the functional dependency graph and generate the inference chains
Maximize data utility
Semantics relatedness detection
A sets of attributes to be partitioned
A sets of attributes to be partitioned
A sets of attributes to be visible together
2_Partioning step
Model the constraints satisfaction as a Hypergraph representation
Partitioning
Secondary nodes
1st secure partition
2nd secure partition
n th secure partiton
Produce violation transactions (Q1, ..., Qn)
User Query
3_Query evaluation
Primary node
Role-Based Access Control
Local queries
Distributed queries
Minimal violation transactions
History-Based Access Control

Figure 7. The proposed methodology to generate secure partitions and lock suspicious queries.

reformulate the problem as a multi-objective function F to be optimized. Therefore, we propose a greedy algorithm to partition the constrained hypergraph into k parts while minimizing the multi-objective function F.

(3) Query evaluation model: In this step we propose a monitor module to mediate every query issued from users against data stored in distributed partitions. The monitor module is built on top of *Apache Spark* system and it contains two mechanisms: a Role-Based Access Control mechanism and History-Based Access Control mechanism.

> The first mechanism checks for user role who issued the query and if it is not granted to execute distributed query then this later will be forwarded to CSP, otherwise the query is forwarded to the History Access Control mechanism which takes as input a set of violating transactions to be prohibited and checks if the cumulative of user past queries and current query could complete a violating transaction. If it is the case, the query is revoked.

6. Other future research and challenges

Since the discussed works are recent, there are a number of concepts associated to access control, privacy, data outsourcing and database semantic which could be considered to ensure better data security and utility in the cloud. Hence, there are other research directions to pursue:

Functional dependencies should be considered as a source of threat in data outsourcing scenarios: unlike the approaches in Turan et al. (2017, 2018), we aim in our future work to prevent inference from occurring in distributed cloud database. Our approach is graph-based that firstly detects inference channels caused by functional dependencies and secondly breaks those channels by exploiting vertical database fragmentation while minimizing dependencies loss.

Authors deal only with semantic constraints represented by functional and probabilistic dependencies as a source of inference. However, other semantic constraints, example inclusion dependencies, join dependencies and multivalued dependencies should be considered as sources of inference.

A further interesting direction is when the workload will become available after the database is set up in the cloud: the challenge is how to dynamically reallocate the distributed database fragments among distributed servers while retaining access control policy?

7. Conclusion

We gave in this paper a literature review of current and emerging research on privacy and confidentiality concerns in data outsourcing. We have introduced different research efforts to ensure users privacy in Database-as-a-Service paradigm with both communicating and non-communicating servers. Besides, an insightful discussion about inference control was introduced. We also pinpoint potential issues that are still unresolved. These issues are expected to be addressed in future work.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributor

Adel Jebali is a PhD student in computer science in Tunis El Manar University-Tunisia and Actually member in Electronic Systems and Communications Networks Laboratory. He is also a technologist professor in ESPRIT School of Engineering-Tunisia.

Salma Sassi received her PhD in Computer Science from the University of INSA DE LYON-FRANCE in 2009. She is currently an assistant Professor in the Computer Science Department at the university of

Jendouba in Tunisia. She is the head of DocSys Team (http://www.fsjegj.rnu.tn/Fr/equipes_11_824). She is member of OpenCEMS industrial Chair (https://opencems.sigappfr.org/). His current research interests are in the areas of Data management and Data semantics. Salma Sassi has published in international journals, books, and conferences, and has served on the program committees of several international conferences and journals.

Abderrazak Jemai, Professor at National Institute of Applied Science and Technology (INSAT), University of Carthage, Tunis, Tunisia. He received an Engineer degree from the University of Tunis (ENSI), Tunisia in 1988 and the DEA and "Doctor" degrees from the University of Grenoble (ENSIMAG-INPG), France, in 1989 and 1992, respectively, and he received his Habilitation Degree in 2012, all in computer science. He became Full Professor in computer Science in July 2018. From 1989 to 1992 he prepared his thesis on simulation of RISC processors and parallel architectures. Since 1993, his interests are focused on high level synthesis and simulation at behavioral and system levels within AMICAL and COSMOS at TIMA Laboratory in Grenoble. Abderrazak Jemai became an Assistant at the ENSI University in Tunis in 1993 and an Assistant-Professor at the INSAT University in Tunis from 1994 to 2013. In 2013, he became an Associate- Professor in Computer Science at INSAT Institute and Full Professor in computer Science in July 2018.

References

Aggarwal, G., Bawa, M., Ganesan, P., Garcia-Molina, H., Kenthapadi, K., Motwani, R., Srivastava, U., Thomas, D., & Xu, Y. (2005). *Two can keep a secret: A distributed architecture for secure database services*. CIDR.

Alsirhani, A., Bodorik, P., & Sampalli, S. (2017). Improving database security in cloud computing by fragmentation of data. In *2017 International Conference on Computer and Applications (ICCA)* (pp. 43–49). IEEE.

An, X., Jutla, D., & Cercone, N. (2006). Dynamic inference control in privacy preference enforcement. In *Proceedings of the 2006 International Conference on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services* (p. 24). ACM.

Biskup, J., Embley, D. W., & Lochner, J. H. (2008). Reducing inference control to access control for normalized database schemas. *Information Processing Letters*, *106*(1), 8–12. https://doi.org/10.1016/j.ipl.2007.09.007

Biskup, J., Hartmann, S., Link, S., & Lochner, J. H. (2010). Efficient inference control for open relational queries. In *IFIP Annual Conference on Data and Applications Security and Privacy* (pp. 162–176). Springer.

Biskup, J., & Preuß, M. (2013). Database fragmentation with encryption: under which semantic constraints and a priori knowledge can two keep a secret? In *IFIP Annual Conference on Data and Applications Security and Privacy* (pp. 17–32). Springer.

Biskup, J., Preuß, M., & Wiese, L. (2011). On the inference-proofness of database fragmentation satisfying confidentiality constraints. In *International Conference on Information Security* (pp. 246–261). Springer.

Bkakria, A., Cuppens, F., Cuppens-Boulahia, N., & Fernandez, J. M. (2013a). Confidentiality preserving query execution of fragmented outsourced data. In *Information and Communication Technology-EurAsia Conference* (pp. 426–440). Springer.

Bkakria, A., Cuppens, F., Cuppens-Boulahia, N., Fernandez, J. M., & Gross-Amblard, D. (2013b). Preserving multi-relational outsourced databases confidentiality using fragmentation and encryption. *JoWUA*, *4*(2), 39–62. https://doi.org/10.22667/JOWUA.2013.06.31.039

Bollwein, F., & Wiese, L. (2017). Separation of duties for multiple relations in cloud databases as an optimization problem. In *Proceedings of the 21st International Database Engineering & Applications Symposium*. (pp. 98–107). ACM.

Bollwein, F., & Wiese, L. (2018). On the hardness of separation of duties problems for cloud databases. In *International Conference on Trust and Privacy in Digital Business*. (pp. 23–38). Springer.

Brodsky, A., Farkas, C., & Jajodia, S. (2000). Secure databases: Constraints, inference channels, and monitoring disclosures. *IEEE Transactions on Knowledge and Data Engineering*, *12*(6), 900–919. https://doi.org/10.1109/69.895801

Chen, Y., & Chu, W. W. (2006). Database security protection via inference detection. In *International Conference on Intelligence and Security Informatics* (pp. 452–458). Springer.

Ciriani, V., Di Vimercati, S. D. C., Foresti, S., Jajodia, S., Paraboschi, S., & Samarati, P. (2007). Fragmentation and encryption to enforce privacy in data storage. In *European Symposium on Research in Computer Security* (pp. 171–186). Springer.

Ciriani, V., di Vimercati, S. D. C., Foresti, S., Jajodia, S., Paraboschi, S., & Samarati, P. (2009a). Fragmentation design for efficient query execution over sensitive distributed databases. In *2009 29th IEEE International Conference on Distributed Computing Systems* (pp. 32–39). IEEE.

Ciriani, V., Di Vimercati, S. D. C., Foresti, S., Jajodia, S., Paraboschi, S., & Samarati, P. (2009b). Keep a few: Outsourcing data while maintaining confidentiality. In *European Symposium on Research in Computer Security* (pp. 440–455). Springer.

Ciriani, V., Di Vimercati, S. D. C., Foresti, S., Livraga, G., & Samarati, P. (2011). Enforcing confidentiality and data visibility constraints: an obdd approach. In *IFIP Annual Conference on Data and Applications Security and Privacy* (pp. 44–59). Springer.

De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S., & Samarati, P. (2010). Fragments and loose associations: Respecting privacy in data publishing. *Proceedings of the VLDB Endowment*, *3*(1-2), 1370–1381. https://doi.org/10.14778/1920841.1921009

De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S., & Samarati, P. (2016). Efficient integrity checks for join queries in the cloud 1. *Journal of Computer Security*, *24*(3), 347–378. https://doi.org/10.3233/JCS-160545

Delugach, H. S., & Hinke, T. H. (1996). Wizard: A database inference analysis and detection system. *IEEE Transactions on Knowledge and Data Engineering*, *8*(1), 56–66. https://doi.org/10.1109/69.485629

de Mantaras, R. L., & Saina, L. (2004). Inference attacks in peer-to-peer homogeneous distributed data mining. In *ECAI 2004: 16th European Conference on Artificial Intelligence, August 22–27, 2004, Valencia, Spain: Including Prestigious Applicants [sic] of Intelligent Systems (PAIS 2004): Proceedings* (vol. 110, p. 450). IOS Press.

di Vimercati, S. D. C., Foresti, S., Jajodia, S., Livraga, G., Paraboschi, S., & Samarati, P. (2014). Fragmentation in presence of data dependencies. *IEEE Transactions on Dependable and Secure Computing*, *11*(6), 510–523. https://doi.org/10.1109/TDSC.2013.2295798

di Vimercati, S. D. C., Foresti, S., Jajodia, S., Paraboschi, S., & Samarati, P. (2013). Integrity for join queries in the cloud. *IEEE Transactions on Cloud Computing*, *1*(2), 187–200. https://doi.org/10.1109/TCC.2013.18

Farkas, C., & Jajodia, S. (2002). The inference problem: A survey. *ACM SIGKDD Explorations Newsletter*, *4*(2), 6–11. https://doi.org/10.1145/772862.772864

Guarnieri, M., Marinovic, S., & Basin, D. (2017). Securing databases from probabilistic inference. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)* (pp. 343–359). IEEE.

Haddad, M., Stevovic, J., Chiasera, A., Velegrakis, Y., & Hacid, M. S. (2014). Access control for data integration in presence of data dependencies. In *International Conference on Database Systems for Advanced Applications* (pp. 203–217). Springer.

Hinke, T. H., & Delugach, H. S. (1992). Aerie: An inference modeling and detection approach for databases. In *Sixth Working Conference on Database Security* (p. 187).

Hudic, A., Islam, S., Kieseberg, P., & Weippl, E. R. (2012). Data confidentiality using fragmentation in cloud computing. *International Journal of Communication Networks and Distributed Systems*, *1*(3/4), 1.

Katos, V., Vrakas, D., & Katsaros, P. (2011). A framework for access control with inference constraints. In *Computer Software and Applications Conference (COMPSAC), 2011 IEEE 35th Annual* (pp. 289–297). IEEE.

Moher, D., Liberati, A., Tetzlaff, J., Altman, D. G., & Group, P. (2009). Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. *PLoS Medicine*, *6*(7), e1000097. https://doi.org/10.1371/journal.pmed.1000097

Rath, S., Jones, D., Hale, J., & Shenoi, S. (1996). A tool for inference detection and knowledge discovery in databases. In *Database security IX* (pp. 317–332). Springer (1996).

Samarati, P., & Di Vimercati, S. D. C. (2010). Data protection in outsourcing scenarios: Issues and directions. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security* (pp. 1–14). ACM.

Sayah, T., Coquery, E., Thion, R., & Hacid, M. S. (2015). Inference leakage detection for authorization policies over rdf data. In *IFIP Annual Conference on Data and Applications Security and Privacy* (pp. 346–361). Springer.

Sellami, M., Hacid, M. S., & Gammoudi, M. M. (2015). Inference control in data integration systems. In *OTM Confederated International Conferences On the Move to Meaningful Internet Systems* (pp. 285–302). Springer.

Staddon, J. (2003). Dynamic inference control. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery* (pp. 94–100). ACM.

Su, T. A., & Ozsoyoglu, G. (1991). Controlling fd and mvd inferences in multilevel relational database systems. *IEEE Transactions on Knowledge and Data Engineering*, *3*(4), 474–485. https://doi.org/10.1109/69.109108

Thuraisingham, B., Ford, W., Collins, M., & O'Keeffe, J. (1993). Design and implementation of a database inference controller. *Data & Knowledge Engineering*, *11*(3), 271–297. https://doi.org/10.1016/0169-023X(93)90025-K

Turan, U., Toroslu, I. H., & Kantarcioglu, M. (2018). Graph based proactive secure decomposition algorithm for context dependent attribute based inference control problem. arXiv preprint arXiv:1803.00497.

Turan, U., Toroslu, I. H., & Kantarcıoglu, M. (2017). Secure logical schema and decomposition algorithm for proactive context dependent attribute based inference control. *Data & Knowledge Engineering*, *111*, 1–21. https://doi.org/10.1016/j.datak.2017.02.002

Wang, J., Yang, J., Guo, F., & Min, H. (2017). Resist the database intrusion caused by functional dependency. In *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2017 International Conference on* (pp. 54–57). IEEE.

Xu, X., Xiong, L., & Liu, J. (2015). Database fragmentation with confidentiality constraints: A graph search approach. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy* (pp. 263–270). ACM.

Yang, Y., Li, Y., & Deng, R. H. (2007). New paradigm of inference control with trusted computing. In *IFIP Annual Conference on Data and Applications Security and Privacy* (pp. 243–258). Springer.

Yip, R. W., & Levitt, E. (1998). Data level inference detection in database systems. In *Computer Security Foundations Workshop, 1998. Proceedings. 11th IEEE* (pp. 179–189). IEEE.