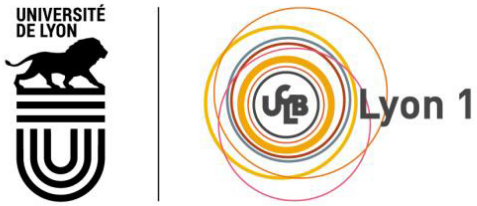
Partage de données tenant compte des politiques de contrôles d'accès

Juba Agoun



N° d'ordre NNT : 2021LYSE1248

THÈSE de DOCTORAT DE L'UNIVERSITÉ DE LYON
Opérée au sein de :
l'Université Claude Bernard Lyon 1

Ecole Doctorale N° 512
Informatique et Mathématique (InfoMaths)

Spécialité de doctorat : Informatique

Soutenue publiquement le 17/12/2021, par :
Juba AGOUN

Data sharing aware of access control policies

Devant le jury composé de :

Nora Boulahia Cuppens Professeure, Université d'ingénierie - Polytechnique Montréal	**Rapporteure**
Laurent D'orazio Professeur, Université de Rennes 1	**Rapporteur/ Président**
Claudia Roncancio Professeure, Institut Polytechnique de Grenoble	**Examinatrice**
Salim Hariri Professeur, Université d'Arizona	**Examinateur**
Hamida Seba Maitre de conférence - HDR, Université Lyon 1	**Examinatrice**
Mohand-Saïd Hacid Professeur, Université Lyon 1	**Directeur de thèse**

To my heroes!

Of course, I mean my family.
You know, for saving my world.

Abstract

With the advent of recent information technologies and the increase in the amount of data, more and more companies and organisations are cooperating through data sharing and exchange for learning and research purposes. To effectively ensure security and privacy, data owners attach a set of rules defined as an access control policy. However, when data is shared between several sources, there may be overlapping data. These redundancies can be a threat when records from the same entity are not considered at the same level of confidentiality. Toward this situation, appropriate filtering of responses to a query must be introduced. Therefore, to ensure data security and confidentiality, each source, having been built independently of the others, defines its own access control policy. The latter provides information that is considered sensitive and therefore not to be disclosed.

In this thesis, we focus on the design and implementation of a framework that allows secure data sharing between two sources. Data sharing is based on the establishment of mappings between entities of two sources. We are interested in using entity matching rules between instances in order to augment the result of queries while ensuring the enforcement of security policies. Furthermore, we seek to bridge the security gap that emerges when two records, from different sources, that represent the same real-world entity are not considered with the same degree of sensitivity.

In this manuscript, we first study the problem of data publishing in the presence of access control rules. We consider the context where a data source is described by a set of publication views and access restriction rules. A view is a table representing a query result intended to be published. The objective is to detect views that leak sensitive information and rather than neutralizing them, we propose a view revision. Our approach uses the necessary and sufficient conditions for a view to comply with a policy request. We formulate a preliminary work that consists of a data-independent method to review views that do not preserve privacy. The goal of this revision process is to strike a balance between data restricting access and data availability.

Subsequently, we propose an entity matching-oriented and policy-oriented methodology to provide a secure data sharing framework. We present an al-

gorithm for translating a query submitted against one schema into an augmented query for the other schema to capture concerning tuples, based on entity matching rules. Then, we provide a methodology to answer queries while maximizing sharing and preserving local access control policies by avoiding any inference leakage that could result from entity matching.

Key words: Data sharing, Entity matching, Record matching, Access control, Query rewriting, Query translation, Data publishing, Data availability.

Résumé

Avec l'avènement des nouvelles technologies de l'information et l'augmentation de la quantité de données, de plus en plus d'entreprises et organisations coopèrent en partageant et échangeant des données à des fins d'apprentissage et de fouille. Afin d'assurer efficacement la sécurité et la confidentialité, les sources attachent aux données un ensemble de règles qu'on définit comme une politique de contrôle d'accès. Par ailleurs, chaque source, ayant été construite indépendamment des autres, définit sa propre politique de contrôle d'accès. Cette dernière, correctement appliquée, désigne les informations qui sont considérées comme sensibles, donc, à ne pas divulguer. Cependant, lorsque les données sont partagées entre plusieurs sources, le chevauchement des données entraîne généralement des redondances liées aux mêmes entités du monde réel. Ces redondances peuvent servir à enrichir les résultats des requêtes mais peuvent aussi être l'origine de menaces. En effet, quand des enregistrements d'une même entité ne sont pas considérés au même niveau de confidentialité, des politiques de sécurité ne sont plus préservées au niveau global. Cette situation renforce l'importance de la gestion de la sécurité dans les systèmes de partage de données.

Ce travail de thèse porte sur la conception et l'implémentation d'un framework qui permet de réaliser l'échange de données fondé sur des correspondances entre instances. En effet, nous nous sommes intéressés à exploiter les règles de matching entre instances afin d'augmenter le résultat d'une requête posée à une source par d'autres résultats à la requête réécrite et exécutée par l'autre source tout en assurant l'application des politiques de sécurité. Une faille peut alors émerger de ce contexte lorsque deux enregistrements, de sources différentes, et qui représentent la même entité du monde réel ne sont pas considérés avec le même degré de sensibilité.

Dans ce manuscrit, nous décrivons dans un premier temps, le problème de la publication de données en présence de règles de contrôle d'accès. Nous considérons la confidentialité dans le contexte de la publication de données en présence d'une instance de base de données, de vues de publication, et de règles de contrôle d'accès. L'objectif est alors d'identifier les vues qui divulguent des informations sensibles et de proposer une révision de celles-ci. Notre approche exploite les conditions nécessaires et suffisantes pour qu'une

vue soit conforme à une politique. Nous formulons un travail préliminaire qui consiste en une méthode indépendante des données pour réviser les vues qui ne préservent pas les règles de contrôle d'accès. L'objectif de ce processus de révision est la recherche d'un équilibre entre la confidentialité des données et leur disponibilité.

Dans un second temps, nous proposons une méthodologie orientée vers la mise en correspondance d'entités et vers les politiques de sécurité afin de fournir un cadre sécurisé de partage de données. Nous présentons un algorithme permettant de traduire une requête posée à une source en une requête augmentée pour l'autre source afin de capturer les tuples concernés, sur la base des règles de correspondance des entités. Ensuite, nous fournissons une méthodologie pour répondre aux requêtes, en favorisant le partage de données et en préservant les politiques locales de contrôle d'accès de chaque source pour éviter toute fuite d'information par une inférence qui pourrait résulter de la mise en correspondance des entités.

Mots-clés: Partage de données, Correspondance d'entités, Contrôle d'accès, Réécriture de requêtes, Traduction de requêtes, Publication de données, Disponibilité des données.

Acknowledgements

Without the support and help of the kind people around me, to only some of whom it is possible to give particular mentions here, this thesis would not have been possible to achieve. I would like to acknowledge my heartfelt gratitude to them.

I want to express my deepest thank you to my thesis director, Pr. Mohand-Saïd Hacid. His patience, thoughtful guidance, and continuous support have been invaluable all along my PhD. As an advisor, you taught me practices that are beneficial for my future academic career. As a friend, if I may say so, I have learned from you how to be resilient in front of failures, deal with pressure, and stay focused on goals. It has been a great honor for me to work under your supervision.

I'm very thankful to all the members of my PhD jury. Prof. Boulahia Cuppens Nora and Prof. D'orazio Laurent for time they spent for reviewing my thesis manuscript, and for their valuable feedback, comments and suggestions. Prof. Hariri Salim, Prof. Roncancio Claudia and Dr. Seba Hamida for devoting their valuable time to review of my thesis.

I express my love and gratitude to my parents, my sister, my brother and my beloved one for their continuous moral support and encouragement with their best wishes. All words of the world cannot describe how lucky I am to have them in my life and your love accompanies me wherever I go.

I especially thank all my laboratory colleagues, whose wonderful presence made it a convivial place to work. I warmly thank all my friends who were always supporting and encouraging me with their best wishes.

List of publications

[AH22] Juba Agoun, Mohand-Said Hacid. *Access control based on entity matching for secure data sharing.* In Service-enabled systems and applications: Privacy Management in Cyberspace. 2022. p. 31-44 (Received: *20 February 2021*- Accepted: *15 August 2021* - Published: *07 January 2022*)

[AH20] Juba Agoun, Mohand-Said Hacid. *Data Publishing: Availability of Data Under Security Policies.* In : International Symposium on Methodologies for Intelligent Systems. Springer, Cham, 2020. p. 277-286.

[AH19] Juba Agoun, Mohand-Said Hacid. *Data sharing in presence of access control policies.* In : OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Springer, Cham, 2019. p. 301-309.

Contents

List of Figures

List of Tables

1

Introduction

Contents

1.1 | Context

In the last few decades, the world has seen an overwhelming speed of digitisation in several domains. The key driver for this jump is due to the increased use of technologies in modern society. With the advent of the Web services, Internet-of-Things (IoT) and Big Data, the size and availability of datasets has increased exponentially, generating an unprecedented amount of information.

The importance of data and the value inherent raises many challenges and brought a considerable number of business opportunities. For example, market-place for collecting huge amounts of consumer information from around the world has emerged, and experts predict a growth prospects CAGR [1] of 6.01% over the

[1] Compound Annual Growth Rate

forecast period[2] (2021–2026). Companies such as Acxiom, CoreLogic and Datalogix are known as data brokers that make a living from collecting and analyzing information then re-sell it for marketing purposes.

Data management systems that combine, link and aggregate information crawled from multiple sectors such as health, insurance, military, human communications and science demonstrated their importance in our everyday life and are still an ever-growing field of study [De 18; BBM15]. Databases remain the core of information systems as they are used as a data container behind an interface. Recently, modern database systems that moved from a internal recordkeeping to a sharing and exchanging over heterogeneous schema found application in various multi-disciplinary fields to explore the high amount of information [Fan+15].

The multi-disciplinary collaboration enables sharing data for a better supporting decision, surveillance assistance, population management, and discovering new insights. Indeed, data sharing projects such as YODA[3] encourages researchers to pursue their investigations after data owners produce the data set. For example, when a clinical trial is conducted for a study, usually the data are not being used after publishing the primary findings. So, instead of keeping the data sets local, providing access to data under policies and specifications is now considered essential to the interpretation of transparency and integrity of the results [KW16].

Data sharing is one of the configurations that allows sources hosting data sets complying with specific schemas to share information. To allow data sharing between the acquainted sources, mappings are necessary and take the form of data-level and schema-level. Data exchange and data integration are two well studied problems, and are based on schema-level as mappings between the sources [KA04]. In a data exchange setting [Fag+05; KMG+17; Kol18], mappings are captured by *source-to-target dependencies* and used to populate a target schema with the data of a source schema. It specifies what source data should appear in the target, and how. In data integration, the approach consists in defining a single entry point to sources by specifying a mapping between the global schema and each source schema. The mapping can be achieved by using one of the well-known approaches, namely, Global As View (GAV) [Cha+94] or Local As View (LAV) [Hal01].

[2]https://www.knowledge-sourcing.com/report/global-data-broker-market
[3]https://yoda.yale.edu/

According to the security organisation Norton, by 2025 there will be more than 21 billion devices [4] reaching 180 zettabytes [5], which will increase the demand for data reporting, analysis and monitoring. Hence, publishing this data produced by businesses and customers, stemming from several sources, raises serious privacy concerns. Information systems hold a crucial amount of data to organisations and businesses that are essential to provide more tailored and personalized services [Aya+14]. Medical conditions, political affinity and ethnic origin are considered to be sensitive, yet highly significant to organisations across a wide range of sectors; they are very sought after by adversaries and malicious organisations.

Using shared information on daily basis puts the data provider in risk to compromise the confidentiality. Users want to believe that manipulation of their information is secure and their privacy is conserved. In consequence, to enable environments with democratized access to data sources meet with legislation and security policies (e.g., GDPR), information system needs guidance to comply and proper functioning.

To address security issues and provide protection guarantees for database applications, both research and practice has studied diversified concepts (e.g., confidentiality, integrity, availability, etc.) and designed approaches (e.g., authentication, access control, encryption, etc.). Protecting the managed data from unauthorized operations is one of the key components of the security infrastructure.

Don et al. [Don+15] identified four primary safety factors for a secure sensitive data sharing:

- Secure data transmission: Security issues raise when sensitive data are transferred from one data owner's to another.
- Access and computing: Some sensitive data could be disclosed from a direct or indirect access. In addition, other sensitive data could leak through or during computing.
- Infrastructure: All the issues that could raise form the physical storage platform.
- Data destruction: there are issues involving secure data destruction. Some research institutions and scholars at home and abroad have made positive contributions to exploration and research aimed at solving these security problems.

[4] https://us.norton.com/internetsecurity-iot-5-predictions-for-the-future-of-iot.html
[5] https://www.statista.com/statistics/871513/worldwide-data-created/

In this thesis we focus on the second factor, *Access control*. There are two basic strategies to protect data from disclosure of sensitive data; Restricting information in the data sources before publication, or restricting access to the data while sharing. The access control is a traditional mechanism with which a system allows or prohibits the actions requested by the users [Fer10; SV00]. Thus, information systems involving data intended for broader use should maximize sharing and availability. Meanwhile, data sharing frameworks should comply with security policies and guarantee strict access to data.

1.2 | Research challenges

In this thesis, we focus on the security challenges that are mainly raised when sharing data between parties. Our goal is to define a methodology for sharing data without compromising the security of any collaborating data source. One big obstacle for parties to data sharing is the storage location. Thus, our hypothesis for such a system is that the data remains local for each source. In this configuration, we assume that the data has a certain level of heterogeneity since sources have been constructed independently. Therefore, a same real-world entity is usually represented differently in two different sources. Furthermore, each source uses its own access control policy to protect its data, different from the others. Our central challenge identified in such a context is: "*How can a query submitted on one data source be allowed to retrieve additional results form another data source without compromising any local access control policies?* "

When data is shared between sources of the same domain, important portions of it are overlapping, due to multi-representation of the same real-world entity. These emerging redundancies serve to enrich query results but they can also cause serious threats, if not taken into account. Indeed, when two records of the same real-world entity are not considered at the same level of confidentiality, security policies are no longer preserved at the global level. Traditional access controls offer an efficient mechanism to protect data locally. However, in such a data sharing architecture (see Figure 1.1), without providing an additional layer of protection, returned results from an external data source could be harmful when they are presented to a user without any pre-processing. Hence, in this research investigation, we are looking at:

- Data Security, policy compliance and information use: Handling sensitive data

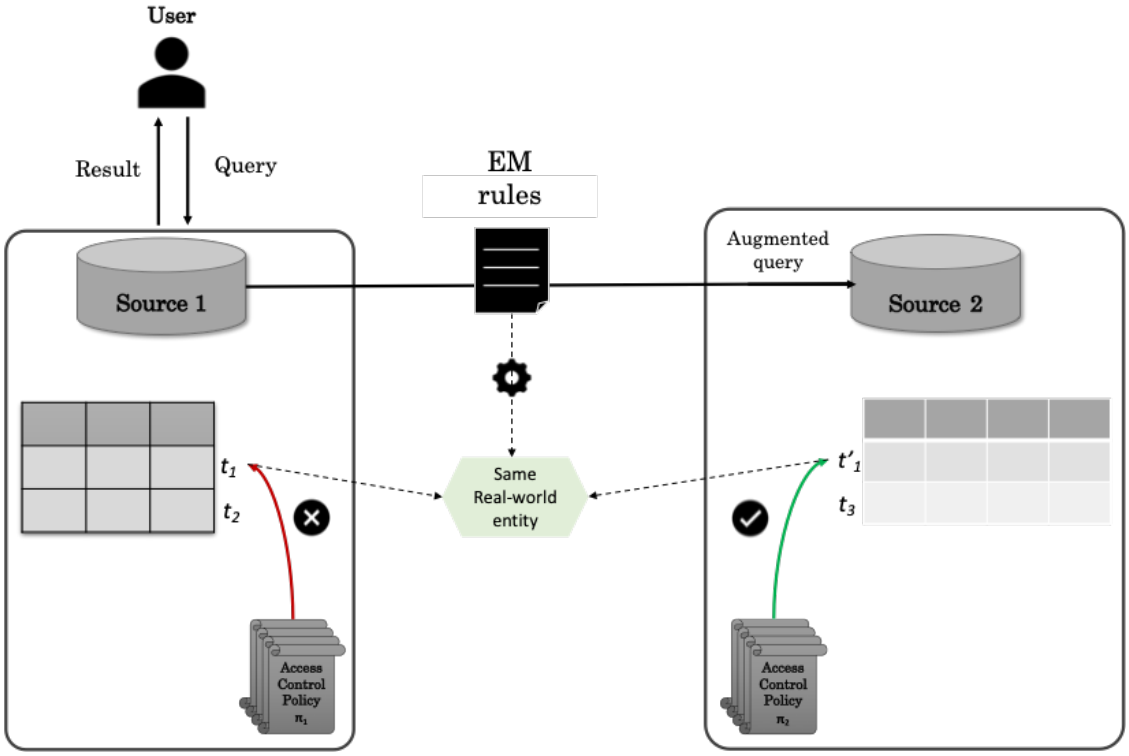


Figure 1.1: Studied data sharing threat

are explicitly being mandated through laws and regulations such as GDPR[6] [God17] and HIPAA[7] [EAH21]. Meanwhile, to provide the public's health and well-being with high-quality health care, the need for sharing data among multiple sources is obvious evidence. However, lawmakers and data owners should be careful henceforth to consider finding a better balance between law restriction and data availability.

- Protection from inside threat: It is important to point out that establishing an access control is not always sufficient to ensure the preservation of designed security policies. Indeed, when a user is authenticated s/he may legally access information supposed to be secret for him/her that would be leaked because of non-harmonised security policies. Thus, it is fundamental to provide a fine-grained top mechanism to our target configuration.

[6]General Data Protection Regulation

[7]Health Insurance Portability and Accountability Act Health Insurance Portability and Accountability Act

1.2.1 | Objectives

Sources collaborating to share data are assumed to be autonomous and with independent schema designs. Similarly, security policies are designed and established separately from each other. The autonomy refers to the fact that they continue to work properly (i.e., update database, add user, and apply new restrictions).

Our key objective is to provide data protection with respect to access control policies that could enforce continuously the secrecy of sensitive information within a data sharing configuration. Information that is considered secret in a database should also be secret when it is shared, and especially if the latter has duplicates or similar in other sources. Indeed, our goal is to offer a data sharing configuration that can enable at least two data sources to enrich their user's query result without violating any restriction rule, especially when a similar information is in different locations.

We investigated these objectives by proceeding as follows: First, we study the different research fields that have addressed this problem and proposed solutions. Second, we analyze the proposed approaches and we explain the best way to combine them. Then, we settle the foundations of our approach and formalise it in the light of the observed threats. Finally, we design a methodology built on relevant concept we identified. The research areas we have explored are:

- Data sharing: There are different ways to combine data steaming from various sources. Our objective in this field is to study different configurations that will enable a user of a given data source to extend their results. Thus, we focus on finding mappings that can rely on approximation without engaging the sources to modify their local design. (i.e., ensuring its autonomy). Furthermore, we studied the problem inside sources to improve the sharing by finding the best configuration for appropriate data publishing.

- Access control: In this field, we study the different access control models that were historically proposed. There are numerous effective mechanisms to secure data against unauthorized actions. In our setting, we need to study the mechanism that maximises sharing for a better balance between restriction and right access. Moreover, it is important to consider that an additional approach is crucial to alleviate the problem that may arise from non-harmonised access control policies at a global level.

- Data matching: This field is an important part of the problem. Indeed, the identification of the same real-world entity allows linking two data sources. Accordingly, our goal is to find a relevant approach, among those existing in the literature, which enables to extend a user query in order to retrieve additional information from external sites. Meanwhile, this entity matching approach would also be beneficial to derive information that could serve to achieve the data protection.

1.2.2 | Contribution

In this thesis, we develop a solution that tackles two sub-problems at different levels. (1) We propose a methodology to address the data publishing problem inside of data sources before sharing; (2) We provide a mechanism to solve the data sharing problem preserving local access control policies at a global level. We summarize the contributions of the thesis as follows:

1.2.2.1 | Data publishing

Before sharing data, database owners build publication views. These views are virtual tables containing result of queries to be exposed later. Publishing data of views without checking the preservation of local security policies is very harmful. Hence, we present a method for detecting and revising privacy violations in this context. It is achieved at design time through a data independent process. This is done relying on two tasks:

1. Identifying views violating access control rules, i.e., presenting high risk for disclosing sensitive information.
2. Propose a revision for views that violates the access control rules to ensure a better data availability.

1.2.2.2 | Data sharing

Once sources agree to publish their data and share it with others, this collaboration needs to set up some necessary parameters. Based on a trusted third party, we propose an approach and its implementation that takes as input local access control policy rules, the entity matching rules, and a submitted query to a given source.

Then, we show how to extend the results of a query over the initial source with the result of the (modified) query over the other sources without disclosing any sensitive information of involved data sources. We show how to address the flaw that arises from overlapping information.

This contribution includes a set of transformation and verification steps to restrict the answer to queries over a data sharing system, where data owners have complete control of their own data. We formalize the concept of query answering based on entity matching specific to our data sharing context. Moreover, we present a technique for translating queries by exploiting entity matching rules. We design twofold strategies to achieve query answering in a data sharing framework that complies with local access control policies.

1.3 | Outline

The thesis is organized in two parts. The first part provides an overview of research efforts in related areas. The second part presents our contributions. The chapters follows the sequence of reported and it is organized as follows:

- In Chapter 2, we analyse the approaches of decentralised database management. We demystify the popular notions of data interoperability to avoid confusion, as their principles overlap in some contexts.
- In Chapter 3, we introduce entity matching, detailing the data matching process.Then, we present the schema aware and agnostic approaches.
- In Chapter 4, we discuss the security of information systems considering important results in the domain. We identify the related work of access control, inference problem and privacy-preserving in data sharing to highlight the lake of a good trade-off between availability and security.
- In Chapter 5, we discuss the motivations behind our thesis. We formally define our problem and discuss it with respect to the concepts described in the previous chapter.
- In Chapter 6, we address the problem of data publishing. We present an approach to assist data owners in identifying and revising views violating defined security policies. This approach aims to provide a better balance between data restriction and availability.

- In Chapter 7, we describes two strategies intended to achieve secure data sharing complying with local access control policies. The first strategy is considered naive but it prevents the disclosure of confidential information that could leak from overlapping data. The second strategy achieves confidentiality in the same context but enables sources to share data without exposing their access control policy.
- In Chapter 8, we summarize our research contributions and the lessons learned. We also present some limitations and potential future extensions of our solution.

Part I

State of the art

2

Data interoperability

Contents

In this section, we demystify the notion of interoperablity since it is popular to be blend with some other notions. Indeed, the definition of interoperability is not clear, it depends on the context. In general, it is the ability of allowing the exchange of data and services between systems or, in a larger perspective, the exchange of information between administrations, between an administration and a citizen or a company, without involving any particular effort on their part [Fra04]. The closest definition to our context is the capacity that data generated by any source can be properly interpreted and used by all other parties participating to construct a collaborative system [She+10].

Thus, achieving the interoperability is concerned with the feasibility of bridging semantic gaps between information systems. It is critical to consider interoperability from two different perspectives: (i) Data interoperability, *i.e.,* it focuses on modeling the exchanged data (ii) Frameworks interoperability, *i.e.,* technologies that includes communication protocols and languages.

Although it is common to consider the framework as a critical part for system collaboration, nonetheless data interoperability remains a must in the development

of technology supporting the operation. Establishing a consortium in a heterogeneous environment between several data sources requires to have at least a common data model. Thus, it is a way for each party to know how to generate and interpret the data within the community. Data modeling is actually the master piece to deal with the problem of affecting the interaction between data sources. Achieving data interoperability of several sources requires to consider building an integrated environment guided by the distribution of the design data. Developing such an environment takes place at various levels:

- **Conceptual level**: Known also as logical level, it considers the description and the development of a standard data model to minimise the need of translation and gain reusability of the acquired data. At this level, business goals and processes operating at every single source are involved to construct mappings to facilitate the data interpretation. However, the differences between models might lead a problem of semantic interoperability [MPB12]. In order to reduce the existing gap between the sources, semantic data model are defined to express the structure of the interoperable environment and enhance their effectiveness for expressing redundent information. [HM78].
- **Technical level**: It consists in the technology supporting the operation of each single data source involved in the collaboration. It is the physical layer that includes the mechanisms for data transformation and transferring over the communication channels.
- **Data management level**: This level is the last step in providing the necessary mechanisms that would maintain the consistency of the integrated environment [BAK07]. Achieving the maintenance of the data consistency needs to provide a transaction management mechanism to enable the control of concurrent access and access control mechanism for regulating the right access on data to only allowed users at the right time.

We focus on attention semantic interoperability for the rest of this section. Obtaining synchronization and inter-model consistency relies on semantic mapping mechanisms between the several data models. The mappings aim at maintaining a coherent operational alignment between the information stemming from the sources to achieve common goals. The first approach to using such a mapping to manage data across different systems was introduced by [MB81]. The authors describe

their mappings as being close to the semantic data model. Later, the concept of the *federated database* [SL90] introduced the basis of a reference architecture for a distributed data management system. It especially mentions the important role of schema mappings as a basic component in defining the structure. Next, we highlight research works in integrating, exchanging and sharing data which the semantic problem is dealt through mappings design.

2.1 | Data integration

Access data steaming from different data sources is the main issue raised by data integration. Indeed, data integration systems aim to combine data residing at different sources and offer a user a unique entry point [Len02]. Such sources contain real data that represent the same semantic concepts but stored under different syntaxes. Approaches to overcome this semantic heterogeneity have architecture characterized by a global schema and a set of sources. The global schema, known also as a mediation schema, is defined through semantic mappings over the sources.

Several systems were proposed, but, we might distinguish two classes of architectures namely materialized and virtual integration:

- Materialized integration (also called warehousing) defines a global schema and stores data provided by the sources. In this approach, the answers to queries are computed without accessing the sources since they are loaded and materialized into a physical database "*the warehouse*" [LWO01].
- Virtual integration consists in defining a global schema without materializing the data. It ensures that the data remain in the sources. The idea behind answering a user query is to rewrite it into a set of queries that are sent to the relevant sources [DHI12].

The two approaches are complementary [Ber+11]. So, each one is more advantageous than the other depending on the desired integration objectives. The management of the changes in the sources makes the virtual integration better because of less updates frequency, the data are retrieved at run time. In contrast, materialized approach offers a better performance when answering queries. The essence of data integration is mappings. They specify the properties of the sources to be linked with the global schema for querying the data. (*see Figure 2.1*).

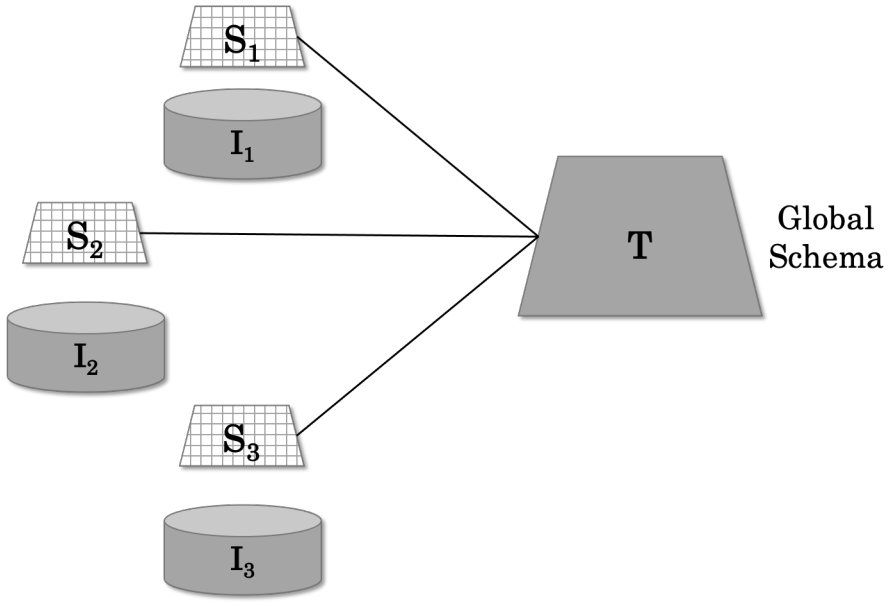


Figure 2.1: The Virtual Data Integration Problem

The logical framework for a data integration system is a triple (G, S, M), where G is the global schema, S is the source schema, and M is the mapping between G and S, made up of a set of assertions relating elements of the global schema with elements of the source schema. As we previously mentioned, the main task in designing data integration is to establish a mapping between the sources and the global schema. There are mainly two basic approaches to virtual data integration (*see* [Len02] *for a survey*):

- **Global-As-View (GAV)**: It requires a global schema to be expressed in terms of the data sources. More precisely, every element of the global schema is associated with a view, i.e., a query, over the sources, so that its meaning is specified in terms of the data residing at the sources.

  Example 1. *Consider the following integration scenario: There are two sources having respectively the following local relations:*

  Source 1 :

  ```
  MovieTitle (mid, Title).
  MovieDetails (mid, year, genre, director).
  ```

  Source 2 :

```
MovieInforamtions (Title, year, genre).
Director (name, movieTitle).
```

The first source has two relations: (i) MovieTitle containing an id of a movie and the title. (ii) MovieDetails with attributes such as the identifier, year, genre and the director. The second data source has also two relations: (i) MovieInformations with Title, year and genre. (ii) Director with attributes such as the name and the title of the movie directed.

These sources are used to build a global relation movie(title, year, genre, director) that consists of a view with four attributes composed by a conjunction of atoms over the source relations.

Global schema :

```
Movie(t, y, g, d) :-- MovieTitle (i, t),
                      MovieDetails (i, y, g, d),
                      MovieInforamtions (t, y, g),
                      Director (d, t).
```

The user directly queries the global schema,in particular the relation Movie. Then, the query is rewritten with source relations to compute and return tuples satisfying the query.

- **Local-As-View (LAV):** The global schema is specified independently from the sources, it is described in the opposite way as in GAV *i.e.,* the global schema is defined first, then the relationships between the global schema and the sources are established by defining every source as a view over mediated schema relation [Ull00].

  Example 2. *Suppose we have a global schema with two relations: (i) Movie(title, year, genre, director) (ii) MovieDetails(title, country, duration, price). Now, if we suppose that we have two sources: (a) S1, containing titles, years and directors of French drama produced after 1980, and (b) S2 containing price, duration of American movies produced after 1990. In LAV, we would describe these sources by the following mappings. Please note that only variables appearing on the right hand sides are assumed to be existentially quantified:*

```
S1(title, year, director) =>
                Movie(title, year, genre, director),
                MovieDetails(title, country, duration, price),
                year > 1980, genre='drama', country='France'.

S2(title, price) =>
                MovieDetails(title, country, duration, price),
                year > 1990, country='USA'.
```

Answering queries in LAV approach is a little bit tricky due to query reformulation (rewriting process). It is not always possible to unfold the definition of the relation in global schema.

The two previous approaches have different behaviours for achieving the same goal of defining a global schema. Query rewriting in GAV is simple but this approach suffers from schema updating; for example, when a source edits its schema or a new source is added to the global schema the whole mapping will be modified. In the other side, LAV does not require any modification in case of any local or global schema update. Nonetheless, to perform query rewriting in LAV there are complex and time consuming algorithms described in [Hal01]. However, to overcome some of the drawbacks of both GAV and LAV, authors of [FLM+99] proposed another class of mapping, named Global and Local As View (GLAV). It is a sort of a super-set of GAV and LAV where we can find the two different mapping logics. This class of schema mappings have been investigated in data exchange [TK09].

2.2 | Data exchange

Data Exchange is the problem of taking data structured under a source schema then create and transform it to an instance of a target schema in such a way to reflect the source data as accurately as possible [KMG+17]. Unlike data integration, in data exchange, the data at the target instance are materialized [Kol05].

Data exchange is concerned with the transfer of data between databases with different schemas, governed by source-to-target tuple-generated dependencies (*s-t tgds*). The formal definition of *st tgd* would be an embedded dependency, in the form of first order logic, in which the right and the left-hand sides are a conjunction

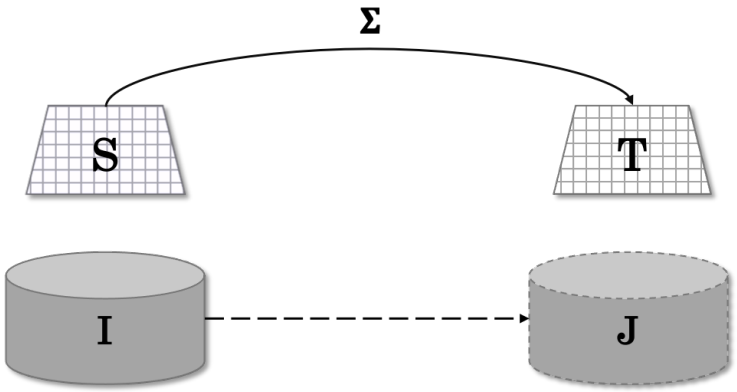


Figure 2.2: The Data Exchange Problem

of relational atoms [1]. The source and target schemas together with the sets of dependencies constitute the schema mapping (*see Figure 2.2*).

The data exchange problem associated with a schema mapping M, corresponding to the triple (S, T, Σ), is the task of constructing a target instance J under a target schema T from an instance source I under a schema S. The target instance J is a translation result of attribute values of I plus some newly invented labeled nulls, such that all of the source-to-target dependencies Σ are satisfied. Such a target instance J is called a solution to the data exchange problem.

Example 3. *Consider the source with the relation Unit(course, lecturer, tutor) and the instance I={(python, Tim, Tim)} i.e., Tim gives both the lecture and the tutorial on python programming, and the target Faculty(idf, name), Course(idc, course) and Teaches(idFaculty, id course) and consider the source-to-target :*

$$\begin{aligned} Unit(C, L, T) \quad \rightarrow \quad & \exists Idc, Idt, Idl \quad Course(Idc, C), \\ & Faculty(Idl, L), Teaches(Idl, Idc), \\ & Faculty(Idt, T), Teaches(Idt, Idc). \end{aligned}$$

The following instance is one of the solutions:

```
J = { Course(C1,python),
      Faculty(F1 , Tim), Faculty(F2 , Tim),
```

[1] A relational atom is an atomic formula under the form $R(t_1, ..., t_n)$ where R is relational symbol and $\{t_1, ..., t_n\}$ are terms (i.e., constants, variables or null values).

```
Teaches(F1 , C1), Teaches(F2 , C1)  }
```

Schema mapping is a complex collections of logical statements, especially when the number of sources grows. Hence, this led to address a new problem known as *mapping selection or discovery*. Indeed, it can exist an infinite schema mappings solution fitting data in the sources. Approaches to schema mapping discovery considered a variety of paradigm to select, from a large set of possible mappings, the best one.

Pioneer work in this area is Clio system[Fag+09][MHH00] that introduced query discovery for mapping creation. Initially, query discovery covers the detection of inferences over relational constraints. Later, mapping discovery has been facilitated using other techniques such as inclusion dependencies query discover [TPN17] blend with the use of metadata as query logs mining [EEL11]. There is another complementary approach called example-driven that uses data of sources to describe the mappings. This way of using data examples as a tool presents a real drawback since the authors of [Ale+11] show the impossibility of uniquely characterizing a schema mapping from a finite set of data examples.

Some other approaches involve the user to score views from a set of candidate views and then propose an optimal set of views based on these scores [Bel+13].

These techniques routinely involve logical constraints with schema modeling in a mediator or third party component. Next, we will introduce a peer-to-peer data sharing that is closely related to our thesis work.

2.3 | Data Sharing

In the two previous sections, we, specially, highlighted the fact that systems operate in presence of schemas on which the mappings are established. Here, data sharing is defined as the problem of sharing data in an environment where constraints cannot be placed on the shared sources of data. The sources are:

- Autonomous: Each source is designed independently from the others and local applications continue running after the definition of the sharing.
- Heterogeneous: The sources could have data represented in distinct models, formats, domains, identifiers, etc.

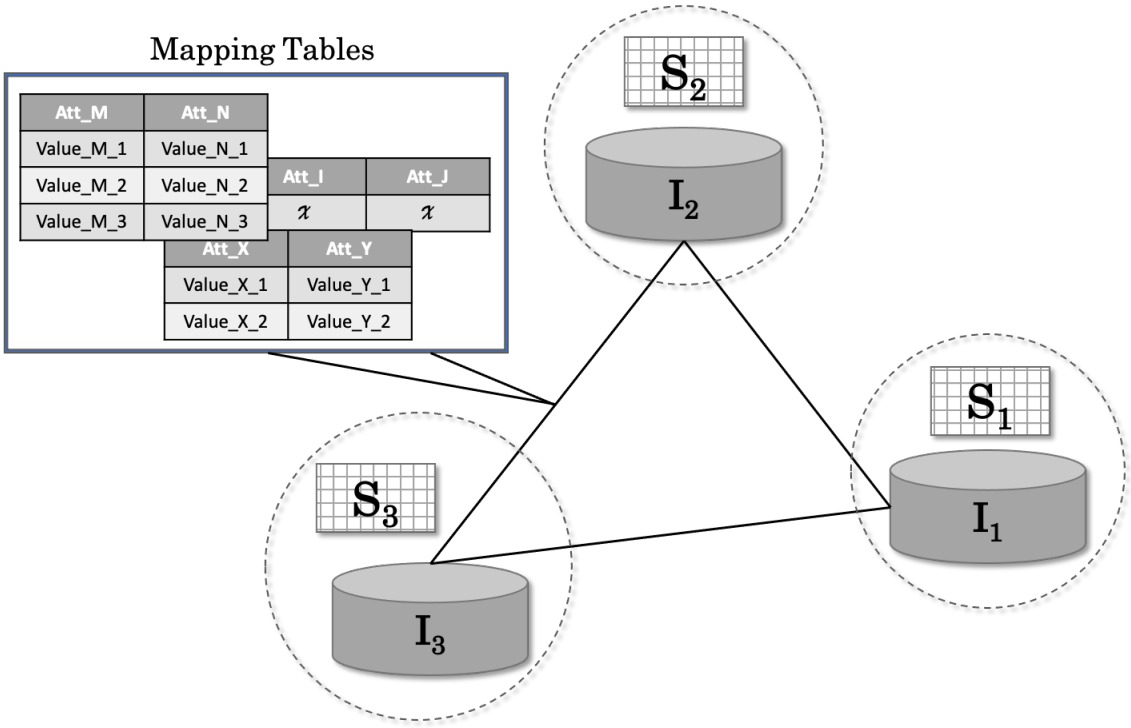


Figure 2.3: Peer-to-peer data sharing probelm

We refer to these autonomous and heterogeneous data sources as peers. Furthermore, data residing in different peers may be closely associated. Indeed, the different peers may use different values to identify or describe the same information (*see Figure 2.3*). Peers share data by defining mappings at a data-level. In literature, this problem is studied under the known name of peer-to-peer systems [KAM03a].

Example 4. *We consider an example in healthcare domain applied to a physician prescribing medications. We suppose two databases namely GreenHospital and NorthLab. The GreenHospital has two relations :*

```
Patient(ohip, name, docRef)
PatientDetails(ohip, test, result, class)
```

Relation Patient has information about every patient received in the hospital with name and the doctor who examined him/her. The relation PatientDetails contains detail about the details of the patient's medical history.

The NorthLab database has only a unique relation for all the patient that visited a laboratory for medical tests. The AdmissionResult relation has the following schema:

```
AdmissionResult(id_admission, name_patient,
                    test_patient, res_patient)
```

The doctor may need to know medical details of his/her patient before prescribing any medications. This information of the patient's medical history (past medication, blood test result, ...) are present not only in the hospital database but also in the database of a medical laboratory.

ohip	name	docRef
364-864	Patricia Lome	Dr. Matkus
963-538	Horor Kime	Dr. Albine
776-231	Piere Laurade	Dr. Fir Menn

(a) Relation *Patient*

ohip	test	result	class
364-864	Hemoglobin	14.6	mmd
364-864	Whitebloodcount	6744	mmd
963-538	Prothrombin	13	meh
776-231	Whitebloodcount	7532	hmm

(b) Relation *PatientDetails*

Figure 2.4: Instance of *GreenHospital* database

id_admission	name_patient	test_patient	res_patient
CWM-019	Patricia Jeane Lome	hmglbn1	13.9
CWM-058	Horor Kiim	prthb-33	14.03
WXD-001	Kine Bourgeoir	Wbld	6532

Figure 2.5: Instance of relation AdmissionResult

Figure 2.4 and 2.5 shows partial instances of databases of our example. Both of the databases store relatively similar information of patient but in different schemes and vocabularies. For instance, Hemoglobin in GreenHospital and hmglbn1 in NorthLab refer exactly to the same test. Furthermore, it is possible to distinguish association between some entities of patient that represents the same person.

To allow sharing data in this peer-to-peer system we rely on data to construct the mappings. To do so, mapping tables are fitting structures to represent how values in different vocabularies may correspond from one source to another [KAM03b].

Figure 2.6 shows two mapping tables that allows to correspond the identifiers and blood test naming of GreenHospital to NorthLaboratory. With such structure, we can provide users to query the system and retrieve all related data.

ohip	id_admission
364-864	CWM-019
963-538	CWM-058

(a) Mapping *ohip-id_admission*

test	test_patient
Hemoglobin	hmglbn1
Whitebloodcount	Wbld
Prothrombin	prthb-33

(b) Mapping *test-test_patient*

Figure 2.6: Mapping tables

Maintaining the mapping tables involves both discovering new mappings and updating the values of mapping stored in the peers. Andritsos et al. [And+04] presented an automated tool for discovering or suggesting associations among data values to enrich the mapping tables. To discover new mappings, their tool rely on characterizing duplication through measuring similarity between records. It is based on de-duplication and record linkage approaches to augment the records of the existing mapping tables.

There have been a number of approaches to data sharing between autonomous and heterogeneous data sources which address the issue by translating queries. Kementsietsidis *et al.* [KA04] considered the problem of data sharing between autonomous data sources and proposed a framework that operates in the absence of schema-level mappings. There, the query translation is computed based on mapping tables that associate data from one source to anther. The authors Ng *et al.* [Ng+03], presented a query translation based on descriptive keywords[2] to associate the schema elements of the sources. The main limitation of this approach is the assumption of consistently using the keywords throughout all participating sources. Closer to our data sharing settings is polymorphic queries [LF11]. It maintains the mappings between the peers based on *contextual foreign keys* (CFKs). CFKs are an extension of foreign keys that make reference to primary keys, by incorporating patterns of semantically related data values. The approach is a query model for *peer to peer* (P2P) systems and consists in explicitly retrieving attributes even when they are not defined at the local peer. For a given query Q_0 in a local peer P_0, Q_0 is evaluated locally. Then, Q_0 is translated to the other peers to conduct horizontal and vertical expansions based on CFKs. The identification of tuples representing the same real-world objects is possible through *matching keys* (MKs), which are same as our entity matching rules. Horizontal expansion extend tuples retrieved from the local peer P_0 by including relevant attributes found in the other peers of the P2P

[2]These keywords are provided by the users to serve a kind of synonyms table

system. Vertical expansion consists in evaluating the query Q_0 in other peers, and find new tuples missing from the local. Our work considers entity matching rules for mapping between sources.

The schema alignment is an important task for data sharing [RB01]. It is a well studied problem in schema matching. Schema alignment allows two schemas to produces a mapping between them to semantically link attributes to each other. Previous work such as in [Ng+03; KA04; LF11] work under the assumption that the constant values of the records can be mapped and normalized. However, databases content (like product names, personal names, place names ...) are, in many cases, semantically heterogeneous. The lack of one unique global domain in such a data sharing system hinders operations like join across data sources. Wiliam Cohen [Coh98] proposed a database logic toward sharing data called WHIRL. This approach extends convetional query languages with some properties of Information Retrieval. The database SQL query supports precise semantic. WHRIL assumes that the content of sources is in natural language text. Under this assumption, WHIRL reasons on text similarity by executing queries and join operations over the data sources through similarity literal. Similarity literal affords WHIRL to support similarity joins across several data sources [Gra+01]. Thus, similarity join operation retrieve information through two values present in different databases that refer to the same real-world entity. In the next section (*section 3*), we will extend in detail the notion of entity matching, important in our thesis.

In our work, we define *similarity predicate* to augment queries to return additional tuples when sharing data. This concept is different from what is known as we "*uncertain predicate*" described in information retrieval [DS07]. Users formulate queries where sometimes a single misspelling of a constant in the WHERE clause leads to an empty set results. To deal with user frustration, uncertain predicate are applied, depending the distance metric, to outputs a probability score of match between constant in the WHERE clause and the attribute values in the database.

In this section, we described how mappings at data-level afford integrating data from several sources. We notice that WHIRL allows the discovery of entities referring to a same real world-object from different sources. Next, we introduce the entity matching concepts and discuss the literature review.

3

Entity matching

Contents

Data matching, record or data linkage and entity matching, are the prominent names used to describe an essential task for identifying common entities stored in several data sources and different formats.

Early application domains to be interested in entity matching were healthcare and censuses [Dun46]. Traditionally, health institutions wanted to link patient history information collected by doctors, hospitals and pharmacies for epidemiological researches. For instance, matching patient information like birth, death, address

of several hospital data with spatial data led the Oxford Record Linkage Study to discover correlations between disease, environmental and socio-economic factors [Gil01].

In the past decades, an increasing number of application domains gained interest in entity matching because of the recent and significant advances achieved in several aspects of the data matching process. The wide use of personal entities for record matching enhanced several application areas (e.g., National security, business, e-commerce, social science ...) to overcome their encountered challenges. For example, to effectively identify fraud, crime, or terrorism, national security agencies and crime investigators rely on a data matching process to aggregate data provided through law enforcement, Internet service providers and financial institutions [Vat+17].

The lack of a common identifier of an entity located in disparate databases is a major hindrance in data management, especially when it comes to data integration. Alternatively, when the keys are distinctly available the data matching may be implemented simply as an exact database join. A particular case of entity matching is when applied on a single data source, this process is called *deduplication*. To improve the quality of data, database owners use the deduplication technique which identifies the matches (i.e., pairs of records that refer to the same real-world entity) either to merge them into a clean representative record or delete the less informative one.

It is important to highlight the difference between schema matching and data matching; in fact, schema matching is the problem of generating mappings between attributes of two schemas. Furthermore, data matching is used as a tool to discover the matches between the conceptual structures (e.g., ontologies, XML schemas ...). In this thesis we will focus on data matching and assume that the attributes are aligned, in other words, we assume that the matches between the elements of schemas are provided as initial information.

The main difficulty in entity matching comes when trying to locate the matches. The level of difficulty varies depending on the structure, format and the content of data. Several types of errors are due to typing, misspellings or lack of information. For instance, we might encounter a typographical error due to phonetic transcription (e.g., "*James*" Vs "*Yames*") or format encoding (e.g.,"*14/03/1994*" Vs "*14031994*"). Thus, using exact comparison to compare values between two records has no sense in this situation. However, the core technique for any entity matching comparison is similarity function, which rather than returning a binary value it measures the

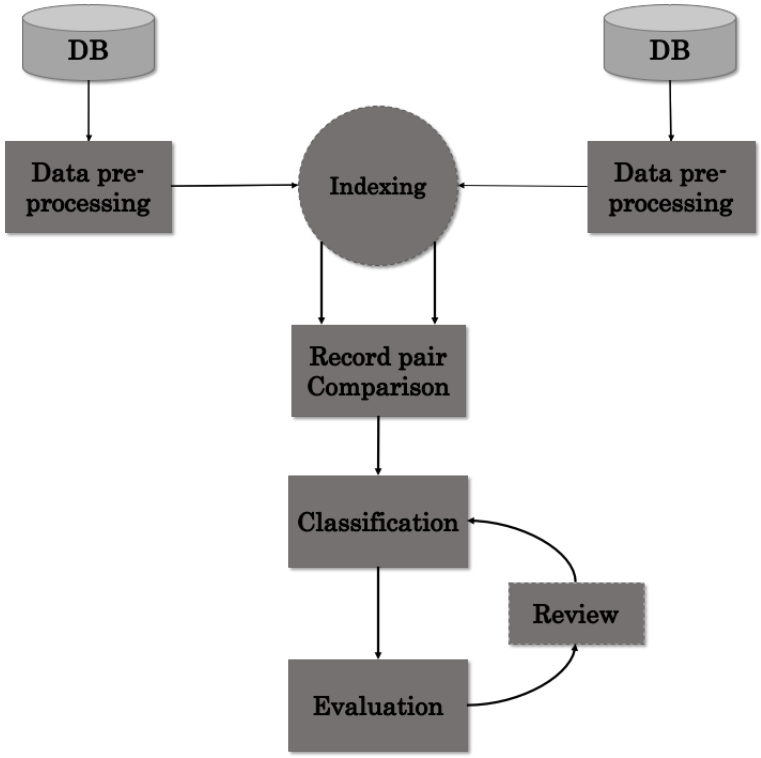


Figure 3.1: Entity matching process [Chr12]

similarity between two values. Further in this manuscript we will provide formal definition of similarity function.

3.1 | Data matching process

In this section, we describe the overview process of data matching with its important tasks illustrated in Figure 3.1.

3.1.1 | Data-processing

Real-world data present in datasets are often incomplete, contain errors and incorrectly formatted. Because of the noise and format's inconsistencies influencing data quality [Cla04; HSW07], data-processing is an unavoidable to clean and standardize the data before record matching [Chu+02]. Thus, for a successful entity matching or deduplication major tasks in data-processing are: (i) Clean characters and words *i.e., remove the unwanted characters (coma, hases, quotes ...) and remove irrel-*

evant words such as stop words [WS92] (ii) Correct misspellings *i.e., standardize the values by reducing name variations and extending abbreviations.* (iii) Segment attributes into a well-defined attribute *i.e., split a value of an attribute that contains several information (e.g., full name) into several values and it is commonly known as parsing.* Recent approach to data-cleaning are involving language grammar [AK09] and semantic associations [Guo+09] to reason about data and address standardisation.

3.1.2 | Indexing

After cleaning and standardizing the data of involved databases, a record of one database needs to be compared with a record of another database to measure their similarity. The naive solution is to apply a Cartesian product between the records of one database and the other database. This naive approach to generate pairs of candidate records leads to a quadratic number. Thus, pair comparisons is time consuming. However, to lower the impact of comparison time , techniques of indexing that dismiss dissimilar pairs have been proposed [Chr11]. Hence, the generated candidate pairs present a higher probability to be a match and will be compared in more detail thereafter. The blocking technique is a traditional approach to indexing [Ste+14]. Basically, it constructs partitions by splitting the databases into non-overlapping blocks. Record of same one block agree on some picked value of a field such as year of birth, gender or zip code. Therefore, records from the two databases with same blocking key are compared with each other. Other advanced approaches to blocking have been explored such as canopy clustering [MNU00], nearest neighbor-hood clustering [VC13] and locality-sensitive hashing [WYP10; Lia+14].

3.1.3 | Record pair comparison

Even with sophisticated achievement of data processing perfect standardization would not be feasible. Mostly, data concern information related to names, address, bibliographic naming, product description that could not be converted exactly same form for all attributes of the databases. Therefore, it is important to employ comparison function to calculate the numerical similarity between attribute values rather than exact comparison [Chr12].

Generated pairs of the previous task require more detailed comparisons i.e., similarity between two record is indicated by comparing several attributes. A numerical vector is obtained for each record pair called *comparison vector*. It is the basis of all data matching techniques, traditional approaches sum all the values of the comparison vector whereas advanced ones involve schema weightings depending if the attribute is more informative [Min06; BG07].

One difficult question to answer in practice is which of the many similarity functions should be suited for each attribute. Indeed, different similarity functions are required in this case, some are specific (data, time, locations...) and some are widely used in approximating string comparison [HD80; Sna07; NH10].

3.1.4 | Classification

The core task of entity matching is classification. The general idea is that the more similar two records are, the more likely they are to match. Classification gather the pairs of records into classes based on the comparison vector [Chr12]. The first class contains pairs of records that refers to same real-world object, this class is called *match-class*. The second class is of the *non-match-class*, it gathers all the pairs that are dissimilar. For example, all the pairs removed by indexing task are implicitly classified as non-match. There are approaches such as probabilistic record linkage [HSW07] that consider a third class, called *potential-match-class*. The pairs that need external knowledge (e.g., from clerical review) are brought together to be humanly reclassified into the two previous classes [HSW07; Pan+17].

Classification approaches can be either training-based or without training [KR10]. In training-based approaches, a set of pair examples are used to train the classifier, each in which are known matching and non-matching records. In the other hand, approaches without training classify pairs of records based on the similarity for direct attributes values [LCW07; Ben+09; Pan+15].

Traditional approaches tend to rank pairs independently of each other, focusing only on the comparison vector. Recent research, on the other hand, aims to exploit collective classification methods [BG07; RDG11]. Indeed, rather than considering only pairwise similarity, these approaches exploit additional information that could be inferred on how the records are related between them. Although collective entity matching outperforms the conventional approaches in terms of accuracy [RDG11], it suffers from its higher computation complexity, which limits its scalability [Chr12;

RDG11]. We will discuss, later in this chapter, two categories of entity resolution, categorized regarding the adopted classification approach.

3.1.5 | Evaluation of quality and preforming the process

The main objectives of developing an advanced new approach for data matching are to achieve high quality and out perform time processing [Chr11]. After record comparison and classification, this task measures the quality of data matching. It refers to how many of the classified matches correspond to true real-world entities. Parameters to measure the matching quality are precision and recall, referred as accuracy, used in several other fields such as data mining, machine learning, information retrieval, etc. All the steps of entity matching process affects the accuracy, the indexing step directly impacts the completeness of data matching since some pairs of records were implicitly classified as non-match without being compared [Chr12].

To evaluate the process of data matching it is important to have a reference of record pairs called *ground-truth* or *gold standard*. It is difficult to acquire such pairs in many application areas because of legislation and the ground-truth data generation process [CG07]. The gold standard data contain true matches (pairs of records that refer to same real world entity), with same characteristics as the studied data, which are manually verified through clerical reviewers.

In spite of all this, manual classification is required to improve the process of classification. It allows approaches with three classes pair-wise classification to decide manually if the pairs of *potential-class* refers to a match or not [GB06]. Some other external reviewing could be also made on parameters. In rule-based entity matching [Pan+17], the maintainability and improvement of matching process is conducted by analysts to refine rules that are necessary for the classification.

The process of reviewing is time consuming and requires a lot of effort, especially when the databases run large number of records. Process of generating ground-truth data faces many issues since it is based on human decision [Chr12]. For example, a manual classification can differ from reviewer to an other depending on her/his expertise, mood and concentration.

We have presented different tasks targeting entity resolution. We recall that the steps remain the same in the case of discovering deduplication.

As mentioned previously, it appears that classification is the core operation of the entity resolution process. Based on the classification we can categorize entity matching into two families depending if they rely on a schema or not.

3.2 | Schema-aware

Entity resolution (ER) frameworks that operate on structured data (e.g., relational data) are considered *schema-based* methods. As the data is described using a scheme, entity resolution methods exploit the schema knowledge through a schema matching to map between attributes. In addition, schema knowledge allows domain experts to address the veracity (e.g., inconsistencies, manual data entry errors, noise, etc.) of data with high effectiveness [PIP20]. Schema-based methods rely on aligning attributes of the data-sets manually by human experts or (semi-)automatically [BMR11].

The matchers are algorithms that specify the nature of how the similarity between two entities is computed [Chr+19]. The are two types of matching techniques [KR10]: (i) Attribute-based technique - it examines the records using a similarity function and applies it on the values of a pair of corresponding attributes, then makes a matching decision independently of the other record pairs. (ii) Context-based technique - also known as collective ER, which considers the context or semantic relationships of different entities, employing graph [NH10] and hierarchical clustering [BG07] approaches, for matching decision.

Furthermore, matchers are involved in the classification of pairs. There are many approaches for making the matching decisions within a matcher. We distinguish the following ones:

3.2.1 | Numerical approaches

During the comparison step, a *comparison vector* is generated for each pair of records. Numerical approaches combine the similarity values of attributes between two records (r_i, r_j) then by taking a weighted sum or weighted average of similarity values a decision is made. The native numerical classification is threshold-based

[Chr11], it compares the sum of weighted [1] similarity values of m attributes to a single threshold θ:

$$\begin{aligned} \sum_{k=1}^{m} f_k(r_i, r_j) \times w_k \geq \theta \rightarrow match \\ \sum_{k=1}^{m} f_k(r_i, r_j) \times w_k > \theta \rightarrow non\text{-}match \end{aligned} \tag{3.1}$$

Other approaches, commonly named *probabilistic ER* [For+01; Win02; KR10], consider the distribution of attribute values in the considered databases. Indeed, [FS69] formalized the problem based on conditional probabilities $P(\cdot|\cdot)$ by estimating the ratio for a pair of records (r_i, r_j) being a match to (r_i, r_j) being a non-match. Then, the ratio is compared to thresholds determined by prior error bound on false matches and false non-matches. However, the difficulty of this approach is estimating the accurate error rate.

Furthermore, Bayesian Networks (BN) are employed to explicitly represent dependencies between attributes in order to decide on the classification of candidates pairs of records. They consists of graphical models that contain information about probability relationships between entities of a domain [Win02]. However, they cannot be applied without representative training data. Some works like in [LCW07] used both of labeled and unlabelled data to compute accurately the probability of two XML entities being a match.

3.2.2 | Rule-based approaches

Rule-based classification uses rules formed as logical predicates of match conditions to derive the match decision [Coh00; Cha+07b]. The rules, like the previous methods, classify the candidates records pair into match and non-match. Match condition are individual tests that are threshold conditions defined on the similarity value between two corresponding attributes. Commonly, the rules are in the form of $P \rightarrow match$, where P is the conjunction of similarity functions conditions of the form $f_k(r_i(A_m), r_j(A'_m)) \geq \theta_k$. The classification outcome of a rule assigns a candidate record pair to match class if P is at *True*, otherwise to non-match class. Nevertheless, in some other configurations, rules can classify pairs into match, non-

[1] Weights are assigned according to the importance and power of attributes

match and *potential match*. The pairs in the latter class are manually validated or not by clerical reviewers [Chr11; NH10]. More details about rule-based approach will be provided in this manuscript in *section 7.3.1*. The accuracy of a matching classification is closely related to how the rules are generated. Indeed, a matching-rule gives a high accuracy when it classifies a match record pairs into a match class. It is not possible to assess rule accuracy without trues match status (*see section 3.1.5*). The accuracy of a rule can be impacted by the number condition in the predicate P. The more conditions are evaluated, the more the rule is specific, thus, covers larger candidate record pairs.

Traditionally, the rules are designed by experts based on domain knowledge. Thus, it is extremely time consuming and error-prone for the expert to generate smaller and concise rules. Furthermore, it is shown that the matching time is dominated by computing similarity function values [Ben+09; Pan+17]. However, alternatives works such as [Wan+11b; Pan+17] have been proposed to identify the best similarity functions and thresholds for effectively finding entities and minimize the time required to apply the parameters. In this view, rules are generated from training data that consist of examples with positive and negative matches. Singh et al. [Sin+17] pointed the fact that other approaches to entity resolution are not *interpretable* (*i.e.,* present a non-understandable form for humans), which makes it difficult for analysts to improve and maintain the ER system. Furthermore, they showed through an experimental process that concise and interpretable rules for end-users could achieve comparable results with the state-of-the-art solutions.

3.2.3 | Learning approaches

The learning approaches automate the process and minimize required manual effort. To improve the accuracy of entity matching various machine learning techniques were investigated. Learning approaches are known to be iterative to improve the accuracy of the resulting classification with less manual effort. Determining suitable parameterizations for matching classification could be either supervised or unsupervised, depending on whether it uses training data:

- **Supervised learning classification**: The feature of this classification is the need of training data. The latter are gathered through training-selection process that choose representative elements of data to be matched and exhibit

the variety and distribution of errors observed in practice. Subsequently, the matcher is trained using these data, also called *labelled-data*. Supervised classification approaches follow three important steps : (1) Select adequate supervised technique [HPK11] and train the matcher. (2) Evaluate the model on testing data which must be different from training data to avoid over-fitting, then evaluate its accuracy. When the accuracy is not good, either adjust parameters of the previous step or change the classification technique. (3) When the minimum required accuracy is reached, the model is applied to the overall data. Two popular supervised techniques are employed in ER. First, *Decision tree*: it is used in several matching and deduplication projects such as *Active Atlas* [TKM02] and TAILOR [EVE02]. The obtained result is a tree where nodes are similarity predicates and leaf nodes correspond to classes. This technique is better favoured by practitioners because of its easy interperetability and visualisation. In addition, the decision trees could be transformed into rules such in Rule-based classification. Second technique is *Support Vector machine (SVM)* [Chr08]: The idea behind is to map the training data (i.e., the comparison vectors with their the labels) into a multi-dimensional vector space. Records pairs of the same class tend to be closer to each other but the gap between classes is wide . The goal is then to find the optimal hyper-plane in n-dimension (n is the number of attributes) with the widest margin between the two classes. MARLIN [BM03] developed a two steps approach that trains an SVM to obtain an adaptive string similarity and as a binary classifier. The first step training, where the focus is on learning the distance metrics and the binary classifier. In the second step, the binary classifier combined with similarly metrics are applied to detect the matching records.

- **Unsupervised learning classification**: Developed as an alternative to training-based approaches, it classifies the records pairs without having a prior information about positive or negative examples. This category is mainly performed by applying clustering-based approach. *Clustering* aims in grouping into clusters records that refer to the same entity. Being very suitable for duplication systems, clustering was widely adopted [Mon00; CGM05; HM09] in identifying objects that refer to the same entity in one single database. Unsupervised learning classification were applied in a few situations that were extremely favorable.

The last generation of learning approaches are named *Active learning* aims to achieve high recall by using a small amount of training data. The idea behind active learning is to rely on user experience in addition to few labeled examples. An active learning model has been proposed [AGK10]. It is based on an interactive process and with a small training data set. The user is asked to specify the minimum precision of the classification trained by either with a *decision Tree* or *SVM* algorithm.

3.3 | Schema-agnostic

Traditional approaches are inadequate for most unstructured data (such as Web Data [PP18]).To address the *Variety* aspect of data management systems caused by unprecedented levels of heterogeneity and noise in schemas, an alternative solution has been proposed [PIP20]. Hence, schema-agnostic approaches aim to address the unaligned-schema EM problem. Indeed, schema-aware approaches rely on schema knowledge, provided by the structured data, to derive the mapping between attributes of distinct data sources.

Schema-agnostic configurations are considered *straightforward* because the record are compared regardless of any schema information. The building blocks technique are build directly from the input of entity records. Efthymiou et al. [Eft+19] proposed a non-iterative and parallel framework for ER in the Web of Data based. It exploits similarity metrics both on content and neighbours of entities regardless of schema information. The indexing process builds a disjunctive blocking graph placing entity descriptions in the same block either because they share a common token in their values, or they share a common name. Then, through schema-agnostic matching rules, the graph of candidates is processed. In [TSS20], authors are propose a schema-agnostic entity matching on structured data. The presented solution proceeds to the concatenation of all attribute values of records and builds sentence-pairs to be treated as a classification problem in natural language processing (NLP) [Dev+18]. The last generations of entity resolution address the Velocity of data, especially as the flow of data increases over shorter periods. To be able to efficiently handle the massive amount of data, schema-agnostic progressive entity resolution methods such as [Sim+18] has shown an important interest. Schema-agnostic progressive ER aims without relying on schema information to identify matches from large datasets where time and/or computational available resources are limited. As

a result, a partial solution with only the best relevant matches is provided in contrast to previous approaches.

4

Data sharing security

Contents

Security is vital practice in data manipulation. Roughly speaking, it concerns all issues related to unauthorized access, piracy and leakage of exchanged data within an information system. Data is a key factor for the stability and sustainability of any organisation. It has therefore become vital to be aware about the safety and security of information storage. Data security in information systems must meet three concepts [BS05]: confidentiality, integrity, availability, known as CIA triad, see figure 4.1.

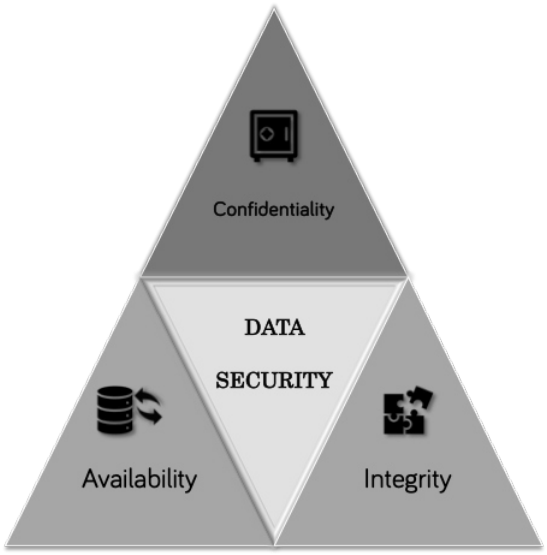


Figure 4.1: CIA triad

Confidentiality

In a closed system, information is viewed only by authorised users of the system. Thus, confidentiality is the duty to ensure no sensitive information is revealed. The concept of confidentiality can be implemented at various levels of any process. However, privacy and confidentiality are two different notions that have been widely blended and mistaken to be the same thing. Confidentiality is similar to privacy, but not the same [And14]. Confidentiality is a crucial aspect for privacy and it is a necessary component. Privacy refers to the right of an individual to be aware and to have control over how his/her personal information is manipulated (collected, stored, shared ...). The term *privacy* is used, in particular, when data are related to personal information.

Availability

The primary concept underlying availability is the ability to obtain data when it is needed. Indeed, collecting data without the capacity to respond to request in a timely manner renders information become unusable. The loss of availability or liveness can occur because of a several breaks in a system (network attacks, power

loss ...) and results into a denial access to data. Such issues caused by an attacker are commonly referred to as a denial of service (DoS) attack [And14]. To prevent form DoS attacks, systems often have redundant components, called fault-tolerance mechanism, so like that if one component fails or experiences a denial, the system ensures availability by switching to the backup module [HFB12; Pat+20].

Integrity

Data is extensively manipulated intentionally or unintentionally when shared and exchanged between and among systems. Therefore, it is a matter of safety for a system that data is accurate. Hence, integrity aims at preventing data from being modified by undesirable users. An unauthorized change or deletion of data or portions of it could be the result of this data modification. Maintaining the integrity in a system requires not only preventing unauthorized changes of data but also need mechanisms able to reverse authorized changes that need to be undone. Such mechanisms are implemented in many applications, such as databases, which allow a roll back of a undesirable transaction so that changes are reversed.

Though, there is neither a single standard decomposition nor a formal understanding for considering these concepts as security properties. Indeed, [BM11] raised the fact of no standard decomposition into confidentiality, integrity, and availability. Furthermore, Jung et al. [Jun+12] showed that in multi-agent systems accountability and non-repudiation, in addition to the CIA triad components, are necessary to guarantee security goals. There exits another decomposition, less known than the previous, named *Parkerian hexad* which is a more complex variation of the CIA triad. The Parkerian hexad encompasses six principles of which three are retaken over the CIA triad with same definition. The variation of Parker [Par98] comes from three additional components : Possession, Authenticity, Utility.

To provide the basic properties of security in an information system, a set of mechanisms as well as techniques are required. According to [SS96] *Authentication*, *Access control* and *Audit* are the foundations for a secure information system. Bertino et al. considered in [BJS95] *encryption* as a necessary requirement in addition to *Authentication* and *Access control* for a suitable system. In recent environments, new requirements, such as trust and reputation, are discussed in terms of goals because some specific factors affect the quality of data. Trust is defined

in [MMH02] as a subjective expectation that an agent has about another's future behavior based on the history of their encounters. However, to design a secure architecture, three essential components are required to prevent, manage and address potential security threats:

- *Authentication*: It is the first interface with exterior and consist of verifying the identities of those using a system or data, usually, by means of a password.
- *Access control*: It is a mechanism for preventing unauthorized actions of authenticated users on data in an information system [Fer10].
- *Encryption*: it is a process of encoding data from its original form into an alternative form known as a ciphertext [BJS95]. It ensures that only intended users will be able to access the data that have been sent, otherwise the ciphertext is unreadable.

The first and the third component are outside of data management scope. Thus, we focus in this section on one of the major components of security, namely on access control. We present, in particular the mechanisms and models of access control. Then, review important research works of the literature that address the security issues in a data sharing setting.

4.1 | Access control

Security in data management systems has generally directed its attention toward access control [Cli+04]. Early works on access control were for relational database systems [GW76; Fag78] and have been influenced by models, originally, developed for protecting resources in Operating System (OS) [Lam74]. Then, much of mechanisms and models have been proposed to meet requirements of new applications (World Wide Web, warehouse, ...), models and environments (distributed, peer-to-peer, stream ...).

Before going further in this section, we introduce important concepts in access control domain. Understanding the following concepts helps to distinguish between the different levels of designing an access control. In other words, the security of system based on an access control is defined from an abstract level to a concrete level as shown in Figure 4.2. It relies on the following concepts:

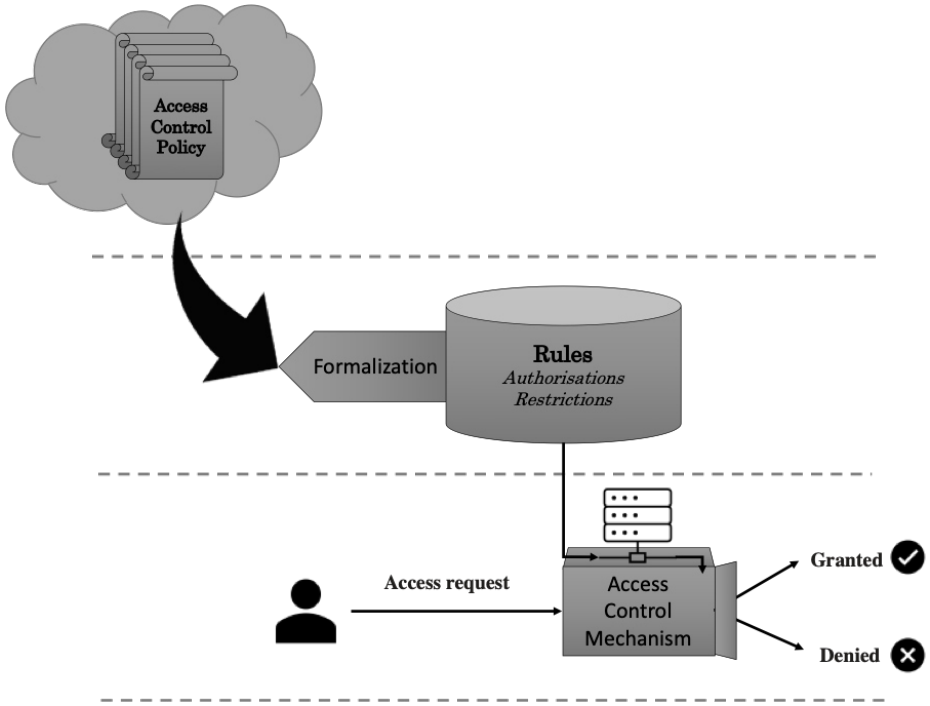


Figure 4.2: Access control security levels

- Security policy: It describes, in an abstract manner, a set of requirements for data protection in a system. At this level, rules are defined to comply with some specification such as laws, regulations, etc. The term policy is used to refer to access authorizations and restrictions to be automatically enforced. Access control policies are considered as high-level guidelines and do not provide any method on how it should be enforced [SV00]. Finding the right compromise on how much information should be accessed or denied is complex. We have two main general principles to deal with this key dimension:
  - **Least privilege.** A high conservative view toward a system since users are only able to access information that are necessary to their legitimate purpose.
  - **Maximized sharing.** Information are shared among the users to satisfy maximum number of access requests, still preserving the confidentiality/integrity of some highly sensitive information.
- Security model: It is the formal representation of the access control. At this level, rules are concretely defined to enforce the authorizations. Access control model provides a framework which describes how security policy is working.

Through the formalization, proof of properties (e.g., security, complexity) are satisfied on the system being designed.

- Security mechanism: Also known as reference monitor. An access control mechanism that implements the security policy and the rules formalized in the model level. The reference monitor consists of a software module that decides whether an access request can be totally or partially authorized [Fer10].

As we can notice, security model is the intermediate level for designing an access control. A security policy captures different requirements of secrecy through practices, regulations, and laws to be enforced. It must also take into account all possible future threats arising from system uses. Several research efforts in the area of access control have focused on developing different strategies. In this vein, it appears that access control can be classified into discretionary, mandatory and role-based access control [Fer10; BS05; SV00]. Historically, discretionary and mandatory were the pioneer access control models and introduced with two important principles [BS05]. The first principle is about the rule formalization. In relational databases, authorizations should be expressed in a logical data model *i.e.,* in terms of relations, attributes, etc. Second principle, databases should allow to decide whether an access to a data item is granted or denied based on some specific conditions. Thereafter, RBAC (Role-Based Access Control) [FKC03] [San98] has been introduced. RBAC conceived for regulating accesses within organizations by associating roles to users and acquire permissions to roles.

4.1.1 | Discretionary Access Control (DAC)

It is called discretionary since users are able to transfer certain of their givers privileges to other users under the regulation of security policy [SV00]. The first approach to discretionary access control was made in [Lam74]. This approach is a conceptual reference model that constructs a matrix called *The access matrix*. More precisely, the columns of the matrix represent objects and the lines represent users. Each cell stores actions that a user could executes on the corresponding object. Identifying the *objects* or data to be protected, the *subjects* that insert, delete and request access to objects, and the actions that can be performed on the objects, and that must be controlled is considered as the first step in the development of such access control

Example 5. *Table 4.1 shows an example of access matrix in a database scenario.*

For instance, in this scenario Lucile and Marc can insert new records into the doctor table. Meanwhile, Lucile is the only user that can select and delete record from doctor table. In contrast, only Marc can insert and select from patient table.

Table / User	patient	doctor
Lucile	-	select, insert, delete
Marc	select, insert	insert

Table 4.1: An example of access matrix

Although the matrix provides a more accurate representation of privileges, it is not effective for the implementation [DS97]. Therefore, the model was refined and formalized by Harrison, Ruzzo, and Ullmann to propose the *HRU model* [HRU76], which aims to analyze the complexity of policy definition. Furthermore, since users have access only to a limited portion of data, storing the matrix as a two-dimensional array is therefore a waste of memory space. Thus, according to [Fer10] and [SV00] two main alternative approaches could be considered for the implementation of real systems:

- **Access control List (ACL):** A list is associated with each object. The list contains all the users and their actions that could be performed on the object. This is the way usually adopted by modern systems [SV00].
- **Capability List:** In this approach, a list is constructed for each user. The list list contains the set of actions that a user can perform on the object. This method is usually suitable for distributed systems, especially, in configurations where subjects can request access to objects hosted by different nodes.

ACL presents an advantage because it allows an immediate check to authorizations holding on an object. Retrieving all the authorizations of a subject is time consuming because it requires the examination of the ACL for all the objects. In contrast, capability list offers an immediate retrieving of subject's privileges but requires examination of all the different capabilities to retrieve the allowed actions for an object.

Discretionary model presents disadvantages for large organizations. When changes become frequent as the number of users increases, it is complex to keep control over the whole system which puts security at risk. Therefore, Mandatory Access Control has been proposed to remedy to these issues.

4.1.2 | Mandatory Access Control (MAC)

In MAC models, accesses to data are regulated based on predefined classifications of users and objects in the system [BS05]. Recall that objects in database management systems are the entities storing the data *e.g., relations, attributes, etc.* and subjects are entities that issue the access request to the objects *e.g., users, services, etc.* The common form of mandatory policy is the multilevel security and the simple way to enforce access control is through defining levels. Thus, objects and subjects are assigned levels which generates *Dominance* relationship because level are ordered. Indeed, MAC models have been widely used in protection of military-oriented environments where access classes follow a certain order. This model is preferred because it aims to control flows between the levels *i.e.,* when information is transferred from one level to another. There are two components:

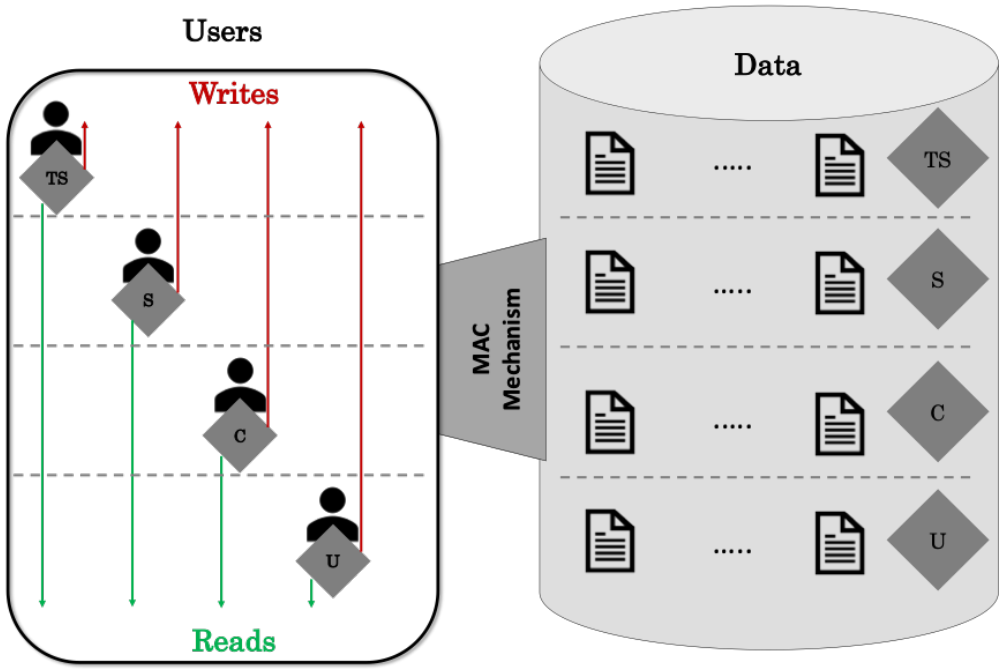


Figure 4.3: Mandatory Access Control model

- **Security level**: Is a hierarchical order labeling. This order follows a certain dominance relationship. For instance, in a set order; Top Secret(TS), Secret (S), Confidential (C), and Unclassified (U), TS dominate S. We denote the domination by $>$. Hence, the general dominance relationship between the levels is : $TS > S > C > U$.

- **Set of categories**: Is a more fine-grained way to determinate dominance. Indeed, set of categories is a unordered set that reflects functional areas (*e.g.,* Administration, Research, and Strategy). The dominance relationship is, therefore, defined in addition the the dominance between levels if the set of categories are included. For example, given two access levels C and U with the dominance $C > U$, a subject with a read access class $(C, \{Research\})$ is not allowed to read objects with access class: $(C, \{Research, Strategy\})$ or $(U, \{Strategy\}$. However, it can read objects with access class $(U, \{Research\})$.

Figure 4.3 illustrates the mandatory access control model which manages the direct access of information flows. Notwithstanding, a classification with the previous components is not enough to prevent from unauthorized move of information from a dominant class to a non-dominant one. For instance, a user with an access class $(TS, \{\})$ can read an information with access class $(TS, \{\})$ then write the obtained information into an object with access class $(U, \{\})$. This example shows a clear indirect disclosure of information, a potential damage resulting when information are moved between levels. Two principles were then introduced by Bell and LaPadula to prevent from this information leakage:

- **No-read-up**: In this principle, a user is authorized to read objects with a lower level than her/his own level.
- **No-write-down** (*property): a user is allowed to write an information only if the level of the information dominates the level of the information. This principle ensures that users do not declassify any information

The multilevel access control has been widely used in commercial data management systems (e.g., Trusted Oracle, Secure Informix ...) [BS05]. The MAC model provides a concrete solution to overcome the information leakage of DAC. In contrast, there is no way to prevent exceptions between the levels. For example, it is not possible to grant access for a user with an access class *Secret* to some part of an object with *Top Secret* level. Therefore, MAC is considered a very rigid model and it is avoided for complex security policies. For this reason, some other proposals such as [CG99] were made but due to the increasing size of information and user management, the non-flexibility of the MAC and DAC, a new model, RBAC, was introduced with new capabilities.

4.1.3 | Role-Based Access Control (RBAC)

Large systems also involve complex security administration when there is a large number of users and exchanged information. For instance, the human efforts to manage the authorization of all subjects and the objects to be protected increase considerably. Thus, the administration of access control becomes a real headache whenever the subject population is highly dynamic which is also not practical because it also increases the risk of inattention, therefore endangering the safety of data.

To simplify the management of access control policies, Role-based Access Control came as an alternative approach to traditional discretionary and mandatory access control. By the early 2000s, [FCK95] introduced the notion of role - a function or position of a user within a given organization, then [Fer+01; San98] proposed a formalization of RBAC fundamentals and defined its components. In RBAC models,

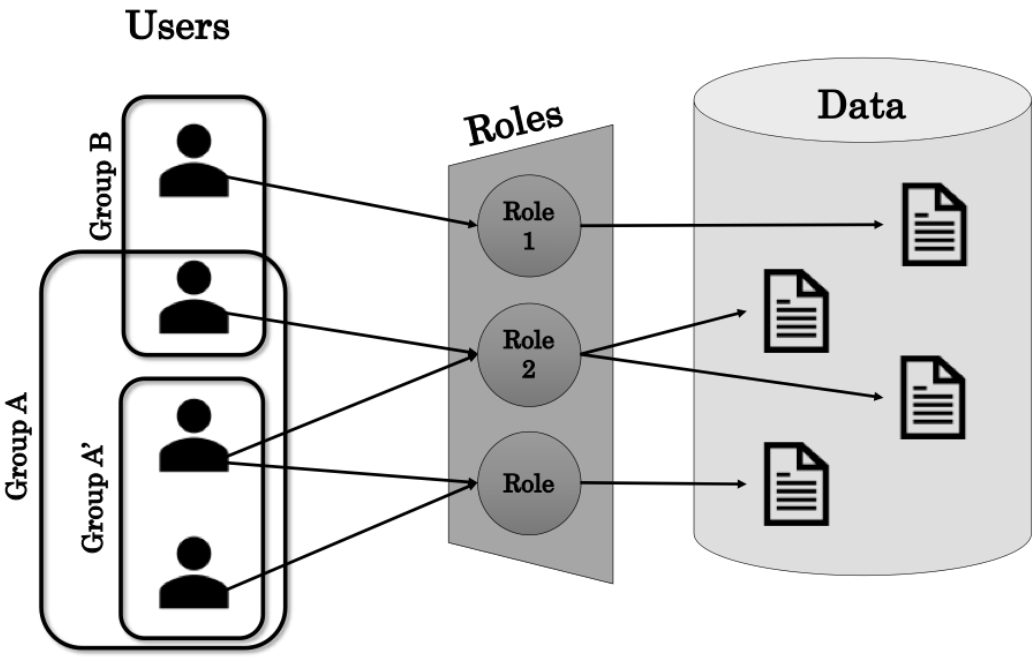


Figure 4.4: Role-based access control

the set of privileges are assigned to a role related to a user rather than directly to his/her identity (see Figure 4.4). A user can acquire an appropriate role between those assigned to him/her depending on the task he wants to execute. The role is now an intermediate level between the subjects and the objects. There is a difference

between roles and groups, it is important not to confuse the two concepts because a role is a set of privileges whereas a group is a set of users [SV00]. Users can enable or disable their role at their discretion when the group membership is always activated and also known. Different concepts have been proposed to enrich Core RBAC *a.k.a.* Flat RBAC, the basic features that any RBAC model should poses. We summarize the most popular features as follows:

- Role hierarchy: The hierarchy is the essential enhancement of RBAC because it fits perfectly with the structure of an organization. This feature allows capturing the line of authority and responsibility within the organisation. When this concept is applied, a role could inherit all privileges of another role.

- Constraint enforcement: Constraints provide RBAC with the possibility of adding constraints to the authorization of a role. This possibility allows expressing further protection for real-world policies. Several types of constraints have been studied and the most investigated type was *Separation of Duties* (SoD) [Cha+07a]. Indeed, SoD aims to limit the privileges given to a user in order to prevent any abuse. For instance, a user can be in possession of two roles and for a given critical task, some policies could state that it needs to be launched by at least two persons, that is, with SoD she/he will not be allowed to perform sensitive tasks alone. Other types of constraints were also proposed such as, cardinality constraints [FBK99] or temporal constraints [Jos+05].

- Least privilege: This feature prevents an intruder to have full control as a legitimate user (e.g., data misuse). It provides a minimum set of privileges needed to perform its tasks within the system.

Role-based access control tries to overcome the limits of DAC and MAC, that is, merging the flexibility of authorizations with additionally the constraints imposed by the organisations. Even though RBAC has been widely used in many real-world applications and systems, it is however a time-consuming model, and still lacks, in terms of flexibility to adapt with changing users, objects, and security policies [Hua+12]. Furthermore, [RJK15] reported the inadequacy of RBAC models when there is a large number of objects because authorisations are over identifiable objects.

4.1.4 | Attribute-Based Access Control (ABAC)

Attribute-based access control (ABAC) [WWJ04] is a flexible approach that can overcome the limitations of RBAC [Hu+15]. It can implement access control policies through computational language and available attributes describing the objects that let it be the ideal in several distributed and ever-changing environments. This model has been also described, in the literature, under the name rule-based access control model [AS04] or credential-based access control in [BS00]. In ABAC model, a subject

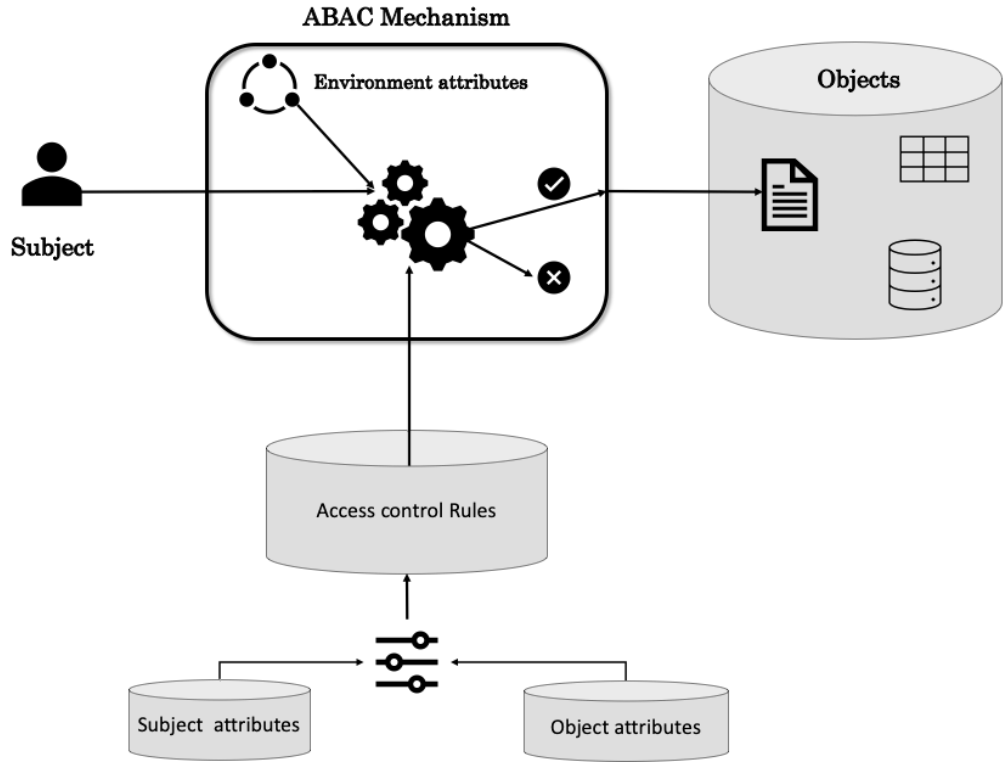


Figure 4.5: Attribute-based access control

(e.g., user) is represented by a set of characteristics about it. Each object or resource (e.g., file, table) has a constraint associated with it. Thus, ABAC offers a better flexibility than the previous model. It is, then, possible to create access control rules without specifying the individual or any relationships that link each subject and each object.

Example 6. *A user, Patrick, is added to a hospital upon hire and is assigned a set of subject attributes, such as Nurse practitioner in the Cardiology department. A folder is created and assigned an attribute, such as Medical Records of Heart*

Patients. The owner of an object adds an access control rule based on attributes of subjects and objects to handles its capabilities. In this respect, a rule is made to limit the access to the folder, such as all Nurse Practitioners in the Cardiology Department can View the Medical Records of Heart Patients. Access decisions are made when a user, such as Patrick, satisfies the access constraints associated with a the folder. (See Figure 4.5)

ABAC offers the ability to handle changes inside an organisation by simply altering attribute values, without requiring modification to the subject/object relationships defining the rules. Hence, ABAC has become popular in open environments that are undergoing continuous change [YT05] and distributed systems [HN10]. This model is introduced as a model that could simulate RBAC and MAC models. For RBAC, the attributes could be considered as labels allowing a subject acquiring a given role, whereas in MAC attributes could allow a user acquiring a security level. ABAC is more complex than RBAC, especially from a policy analysis and reviewing perspective but both models are complementary. Indeed, recent research works such as [Hua+12; RJK15] are exploring the topic of combining the advantages of both RBAC and ABAC models to build a flexible and less complex instance model.

4.1.5 | View-Based Access Control (VBAC)

The previous models are usually considered as coarse-grained access models [Lin+08; Fis+09] even though they provide strong security guarantees. Indeed, they may not be adapted for defining content-centric policy rules, that is, access decisions could depend on the values of the data defined by the policy rules.

View-based approaches offer a general framework to enforce different kinds of policies. It aims at defining access rights of a group, representing zero or more subject which have the same access rights on a portion of data. Indeed, the centeric idea behind VBAC is to define a view representing some piece of data to be prohibited or allowed to be accessed by a subject. The VBAC approach can simulate DAC, MAC, RBAC and even ABAC policies, depending on how the views are defined. View-based access controls are used by databases as an abstract mechanism to enforce policies for available data [Sha+19].

Depending on the system security (open or closed), we refer to a view as an *authorization view* when it grants access to data [Riz+04], otherwise, it is called

secret view when it prohibits access [Qia96].

The view-based access control is in some situations also known as content-based access control. In fact, conventional database access control specifies access rights of each data object (e.g., table) through GRANT and REVOKE instructions. However, such an approach is not always suitable for all policy requirements, especially, when it is related to a specific row in a table. Thus, views offer this fine-grained selection to identify tuples by their contents. View-based access control has been widely applied to relational data models then extended to other data representation models such as XML [Fan+06], RDF [Cho+09] and SPARQL [GL10].

Several fine-grained access control approaches based on views were proposed and could be classified into two categories:

- **VBAC with unconditional query rewriting validity:** The basic idea behind this approach is to maximize the results by rewriting a given query to return an answer containing only authorized data. Indeed, instead of returning an empty set or denying a submitted query Q, the mechanism modifies Q to obtain another query Q' called maximal query (*i.e.,* there is no Q'' such that Q' is included in Q'' and Q" is included in Q). The first propositions made by [Mot89] and [RS00], consists in a rewriting in such a manner that only authorized tuples will be in the answer. The rewriting technique could be viewed as filtering, similarly to the transformation technique applied in semantic query optimisation [CGM90; Gra+00]. Database management systems such as Oracle implemented a mechanism called Virtual Private Database (VPD) [1]. The mechanism appends to each submitted query conditions of access control associated with the relation in the query, thus, ensures that only authorized information will be delivered to the user.
- **VBAC with conditional query rewriting validity:** In this approach, each submitted query is rewritten if the user has the suitable access rights on the whole requested information [Riz+04]. In other words, for a given query Q, the rewriting is performed in such a way to obtain an equivalent query Q' that only uses authorized views to compute the answers to the query. Otherwise, no results are delivered. The interpretation of rewriting is different from the previous approach. This category of VBAC is based on query equivalence defined through query containment [Hal01; PL00].

[1] https://docs.oracle.com/cd/B28359_01/network.111/b28531/vpd.htm#DBSEG98229

The VBAC approaches based on unconditional validity are interesting for their flexibility to enforce a wide variety of access control policies. However, the limitations result from the fact that the executed query is not the same as the one issued by the user, therefore, the returned result may be inconsistent. In contrast, under conditional query rewriting validity the previous drawback does not arise. Nonetheless, conditional query rewriting may become useless if queries are constantly denied.

Access control are mostly designed for protecting systems from direct information disclosure. Therefore, leak of information should arise from indirect requests. This is know as inference problem. In the next section, we introduce the inference problem and describe most important approaches proposed to address this problem.

4.2 | Inference problem

In a database access control context, disclosure of information occurs through an inference when a malicious user can synthesize information supposed to be sensitive from a combination of authorized information [FJ02]. Combining different query results with external information could lead to the inference of sensitive information. Access control presented previously do not prevent form indirect access. Controlling indirect access to data is often referred to as Database Inference Control [SO87]. Although this problem attracted considerable attention and has been studied for many years from different angles, it is still relevant [GMB17].

To be able to infer sensitive information, attackers exploit different sources of external information, such as database schema, data semantics, statistical information and data dependencies (*e.g.,* functional dependencies). Thus, different types of inferences could be listed where for each type specific approaches have been proposed to deal with a particular attack.

Next, we discuss, the different inference types depending on the nature of the attack and the underlying proposed solutions

4.2.1 | Statistical inference

Statistical inference attacks were, historically, the first to be studied. These attacks occur in *statistical databases* [AW89] (*i.e.,* databases that allow to return only aggregated information about data contained within (e.g., mean, count, *etc.*)). Users

are allowed to obtain information about data trend of the elements in the database without access to data of the elements. In those systems, a policy protects a specific portion of information (*e.g.,* salary of an individual) while it provides an aggregate information (e.g., maximum, average and minimum salary ...). An attacker helped with external knowledge combined with some overlapping query results may, for instance, inferring information about individuals supposed to be prohibited from disclosure. Therefore, several solution were proposed mainly categorised into three classes:

- **Query restriction**: It consist on answering queries exactly, but not all queries are permitted. Many methods have been considered in this vein; (1) Query-Set-Size Control - it constrains the number of tuples used to construct query results to avoid that an intruder gets a precise control [Den80]. (2) Query-Set-Overlap Control - It aims to limit the amount of overlap between user's queries and control them in order to stop him/her from compromising the database [DJL79]. (3) Query auditing - Each new query is checked with previous queries and evaluate if the combination between the new and past queries could infer prohibited information [KPR03; SL99].

- **Data perturbation**: Also known as input perturbation queries. There are answered over modified data. Two methods are devoted to avoid that confidential values of an individual be isolated: (1) Probability distribution - Performs data distortion by changing data while keeping the same distribution [LCL85]. (2) Random Data Perturbation (RDP)- It adds random noise to confidential data [LWJ02; OZ10].

- **Output perturbation**: Provides modifications to only query results in contrast to the data perturbation [OS90]. For example, queries are evaluated against a subset of data that is selected randomly instead of the whole database [OS90].

Perturbation technique induces a bias problem, which is a critical drawback since the accuracy of returned values is compromised.

4.2.2 | Semantic inference

Privacy violations through semantic inference occurs in general purpose databases (*e.g.,* multilevel secure databases[2]) [OS90]. A malicious user exploits advantage of semantic constraints (i.e., data dependencies [SO87], database decomposition [HDW97]) that are spawned from database to infer sensitive information via the so-called *inference channel*. The semantic inference problem aims at detecting inference channels which are indirect access that could induce prohibited information disclosure. Mechanisms to deal with this semantic inference depends on its characteristic. Hinke [HDW97] introduced a AERIE (*Activities, entity, relationships Inference Effects*) inference model that represent various types of semantic inferences that could occur in a real database. In [Mor88], a function INFER which is based on information theory characterizes the inference between two objects X and Y. It describes the quantity of information that one would know about Y given the knowledge of X.

Therefore, there have been several works addressing this issue from various viewpoints. We can classify them into three categories [JM95]:

- **Inference by applying constraints on queries**: It occurs by using specific constraints such as join between a classified attribute and an unclassified one. This case points out the vulnerability issue that could arise from labelling attributes with different security levels. However, it was shown that this kind of inference is easily identified and treated by rewriting the queries only with the authorized attributes[MJ88].
- **Inference using metadata**: To infer sensitive information, a malicious user could take advantage from combining returned data from a database with its metadata. Attackers considered *key integrity* earlier, since the relational model was the most common. A key constraint states that a set of attributes is unique for a given relation[Che76], thereafter, it has been generalized into functional dependency.
- **Inference using value constraints**: This kind of attack occurs when it is possible to infer information about the domain of a sensitive attribute [MJ88]. For instance, given two attributes X and Y, where X is labeled as *unclassified*

[2]Data are stored under different security classification and each user is assigned an access right to these data.

and Y as top *secret*. If we assume that the database has the following constraint: $\Delta = X + Y < 100$. Even though Δ does not provide any value of Y, it is still possible for an attacker to guess the values of Y. Indeed, knowing the values taken by X, because X is unclassified, combined with Δ the attacker could infer the values that could be taken by Y.

To detect and deal with the presented inference channels, proposed approaches could be classified into two main categories: Approaches such as [SO87] and [DH96] are said to be Design-time approaches. Whereas, approaches like [BFJ00] are considered as Runtime approaches. A design-time approach improves query execution time but it is too restrictive since it denies a query if it is suspicious. In contrast, a runtime approach maximizes data utility but slows time execution and can be damaging when the number of queries increases.

4.2.3 | Inference channels in data sharing

In [Had+14], the authors proposed a constraint-based access control framework, in the context of integrating several data sources in presence of security policies, to forbid the execution of queries that may lead to disclosure of sensitive attribute associations. The authors proposed a mediator on top of a set of sources and assumes that each source's security policy has been designed independently of other sources. Then, they propose an incremental methodology based on policy revision, able to tackle the inference problem that could arise using semantic constraints (functional dependencies).

Other research works proposed a distributed architecture to ensure no association between attributes in large databases [Cir+09][Vim+18]. It is mainly based on fragmentation intended to keep each part of sensitive information stored in a different fragment. Protection requirements are represented using confidentiality constraints which express restrictions on a single attribute or an association of attributes.

4.3 | Privacy-preserving in data sharing

Data sharing environments are dynamic systems that are constantly evolving and facing numerous breaches. Several works studied security issues that could occur in such a configuration and proposed solutions that could be discussed regarding

this thesis. In [Elm+10], Elmeleegy et al. proposed a privacy-preserving protocol to protect both query results and mappings implemented on top of Hyperion[3] PDMS[Aro+05]. The query answering protocol is based on noise insertion and commutative encryption methods. The challenge was also to ensure privacy-preservation without being unfair to the participating peers in the system *i.e.,* the fairness is to ensure that some peers will not unnecessarily be overcharged among the other peers during the query answering process. Their work did not consider access control policies or any specific security policy.

To preserve privacy concerns in sharing or exchanging data for linkage across different organizations, *Privacy-Preserving Record Linkage* (PPRL) [Vat+17; FSR18] addresses this problem by identifying and linking records that correspond to the same real-world entity across several data sources held by different parties without revealing any sensitive information about these entities. The database owners agree to reveal only selected information about records that have been classified as matches among each other. In [Ina+10], authors proposed an approach that combines differential privacy and cryptographic methods to solve the private record matching problem. In this approach, statistical databases are considered. Differential privacy allows users to interact with the database only through statistical queries. In our case, however, we do not involve any data anonymization technique such as k-anonymity [Swe02] or noise insertion [Dwo08].

Scannapieco *et al.* [Sca+07] proposed a protocol for *PPRL* between two data sources relying on a third-party. Instead of relying on complex cryptographic techniques, it exploits the idea of obtaining privacy by embedding the records of each party in a vector space. Sharing genomic and clinical data was essentially valuable in terms of deriving new insights but presents a high risk for privacy breaching [DLK17]. Access control systems for data sharing were undertaken (e.g., [RM16; Li+18; Don+14]), but unfortunately they do not consider any inference that could result from linked records like in our case. The protocol has two main phases: (1) In phase 1, sources negotiate a secret key, generate mapping expressions design and set embedding parameters between the sources. (2) In phase 2, the sources build their embedding spaces then , they are send to the third-party for matching comparison. Those PPRL techniques do not give any flexibility of specifying which portion of data must be kept as private.

[3]The Hyperion prototype implements the main results/algorithms presented in [KA04; KAM03b]

Secure data sharing was investigated in the area of cloud services [RMV17; LPX20]. Xinhua Dong et al. [Don+15] analyzed security issues involving the entire sensitive data sharing life cycle and proposed a systematic framework for secure sharing of sensitive data on big data platform. It guarantees secure data acquisition, storage, use and destruction based on heterogeneous proxy re-encryption algorithm and through a trusted environment. Hu et al. introduced Ghostor [HKP20], a data sharing system for object stores (which stores unstructured data items and allows shared access to them by multiple users). It provides anonymity through end-to-end encryption and verifiable linearizability based on a blockchain in a threat model. The data sharing system is designed to derive its security from decentralized trust. In this setting, users manage access to their own object stores through Access Control Lists (ACLs). However, the object stores are not linked between them nor even similar. In contrast, Our work assumes that similar data from two sources could reveal sensitive information when they are shared.

5

Problem settings

Contents

We provide an overall view of the problem we consider. First, we present the underlying reasons for our thesis. We see the importance to discuss our approach with respect to the three dimensions described in the previous chapter. Then, we describe the problem of data sharing in presence of security rules problem. Finally, we describe our proposed methodology.

5.1 | Relation between the different (studied) dimensions

In the previous chapters, we presented relevant concepts and approaches that we believe should be considered together when one aims at securing data sharing between multiple parties. We presented the dimension that enables information to be enriched. Subsequently, we have shown the main difference between integration, exchange and sharing of data. We observe that the data sharing configuration is instance-oriented, whereas the others are mainly schema-driven. Data sharing offer the ability to retain data locally and be more flexible for changes.

We discussed the entire entity matching process with the classification of known approaches. We come up with the conclusion that rule-based entity matching is the most reliable for our study problem. Indeed, mapping between data sources guided by data needs a structure that could link between database instances. In this vein, entity matching rules are fit to provide an interpretable description of how elements are similar.

Furthermore, the topic of data security has been studied in-depth to address most of the identified issues. We presented the various access control model with their limitations. We noticed VBAC is more flexible (i.e., could simulate all other models) and offer a finer granularity to control more precise portions of data. We also pointed out that non-harmonized access controls are not efficient to enforce data security in a decentralized environment. Dealing with restrictions induces some loss of data availability. We observed that many concrete methods address data sharing security issues mainly based on data encryption. Even though these existing approaches ensure data privacy through strict restrictions, they do not afford a fair balance between data availability and security restriction.

5.2 | Motivation

Large amount of sensitive information are generated though process of sorting, analysis, and mining [Vat+17; Sha+19]. Several examples could be considered, like; measures of alcohol, sexual behaviors, political opinions, etc. Sensitivity is not the same from one domain to another. Indeed, the notion of *sensitive data* varies de-

pending on context, population, and time. For instance, information about raw material supplier is very sensitive in nuclear area while it is not in textile industry.

Generated data from diverse sectors would be useful for learning new insights when they are assembled. Moreover, for successful learning, it is necessary to be in possession of a maximum amount of information. Hence the emergence of commercial entities called *Databrokers* that collect, process, store and share data with other interested parties. Data sharing offers an attractive way to make an environment rich set of information. Sharing data is also motivated by the fact that it can help to discover the hidden values and link between data. However, collaboration is not always an easy task since sharing data with other parties could lead to expose sensitive contents. Struggling with a lack of collaboration is due to the increasing number of regulations and laws that state how information should be stored and accessed.

We are interested in securing a data sharing system at the access level. Sharing systems offer a convenient way to enrich data, especially, when we consider same entity present in several sources. Indeed, information about a real-world entity present in several data sources is in fact a real motivation for knowledge extraction. In this context, each party enforces its own access managements to their data. Thus, enforcing access control at a higher level is a challenging task when dealing with various security policies. We aim at defining a data sharing framework where it is possible to query multiple sources, enforce local access control and struggle the inference that could occur from the overlapping information. Our aim in this thesis:

- **Querying in data sharing environment:** When a query is posed in a standard database, when information or attributes are not available, the result is either an empty set or an error. In contrast, if a query is submitted in a decentralized environment (e.g., peer2peer system), the query is not rejected (or an empty set is returned) when a piece of information is not available locally. In traditional models, the query is augmented through translation tables which extends the results vertically. However, it is time-consuming and not always possible to maintain the translation table. The challenge is, then, to merge data of all collaborating parties for being able to enrich final query results.
- **Access control:** Sources share their data following local policies that dictate how their users interact with the data. However, while it may be simple enough to enforce locally the underlying policies, it is not so in a decentralized global

environment. Indeed, it is essential to guarantee that the source policies will be preserved in higher level. This means that if a piece of information is not allowed to be accessed locally, such an information or a similar information related to it should not be accessed. This property requires to consider a restrictive access policy decision. Although it seems to be severe, it would be more accurate to guarantee non-violation of access rights. In this vein, our motivation is to provide an enriched query result with more access control guarantees.

- **Entity matching:** Sharing data between autonomous data sources involve the presence of overlapping records. Indeed, the same real-world entity could be stored in several sources but represented with approximate information and in different ways from each other. Entity matching is the process that identify the pair of record referring the a same real-world object. In this context, we are motivated to map the participating data sources through entity matching rules instead of translation tables used in traditional approaches. The challenge is therefore to formalise a data sharing system that can handle the use of the similarity function in order to extend the local results. As previously discussed, when there are two representations of the same real-world object with different access rights, it may lead to an inference. Thus, similar information to a record must be neutralized and kept secret when the information in question is supposed to be hidden from one side.

- **Data availability:** Before sharing data, the owners should be aware of the data they are willing to publish *i.e.,* if there is not any information that could violate the security policy defined locally. To proceed, database administrators often create views that are used to allow access to only authorized portions of the data. However, a human decision may lead to errors and needs verification before publishing data. This step represents a tedious task that checks whereas the views are complying with local security policies. This issue is a real motivation for us to elaborate a process that will assist an administrator to revise her/his views that s/he is willing to publish.

5.2.1 | Security threat scenario

Two health organisations, a hospital and blood bank organisation (e.g., EFS), use an electronic health record (EHR) to manage patients admitted to their services. In the scope of providing better support for the administration of blood in emergency care, delivering the best care protocols or regimens for patients [Qui+19], and for deriving new insights to improve the improved diagnosis, it requires data from both institutions. However, neither of them is willing or even allowed, without any regulation, to provide its database. Therefore, the two organisations agree to share and exchange their data under a set of legal restrictions to control access to sensitive data. The security policies will evolve over time, as will the data, depending on changes to the regulations and the database instance.

A patient X is admitted to a hospital, then for some reason, s/he also went to the blood bank. Subsequently, the same patient X wants to assert her/his right to be forgotten (Art. 17 GDPR[1]) or to keep some information private (Art. 18 Art. 21 GDPR) and makes it only known to the blood bank. In consequence, querying the sharing system could lead to the disclosure of some sensitive information from the hospital.

Access controls at a local level are not efficient in such situations. Thus, with an additional mechanism, querying in such a sharing configuration, where sources have almost the same information, should avoid possible inference problems that might result from the data deduplication.

A more detailed use case will be presented in next chapters to illustrate the threat that might occur.

5.3 | Problem discussion with respect to related work

In this section we discuss the different dimensions related to our studied problem regarding the related work reviewed in the previous chapter. We identified the dimensions; data interoperability, access control, and data matching model. Figure 5.1 shows how our problem is constructed around the different dimensions mentioned in the previous chapter. Hence, for each of the three dimensions we consider a

[1]https://gdpr-info.eu/

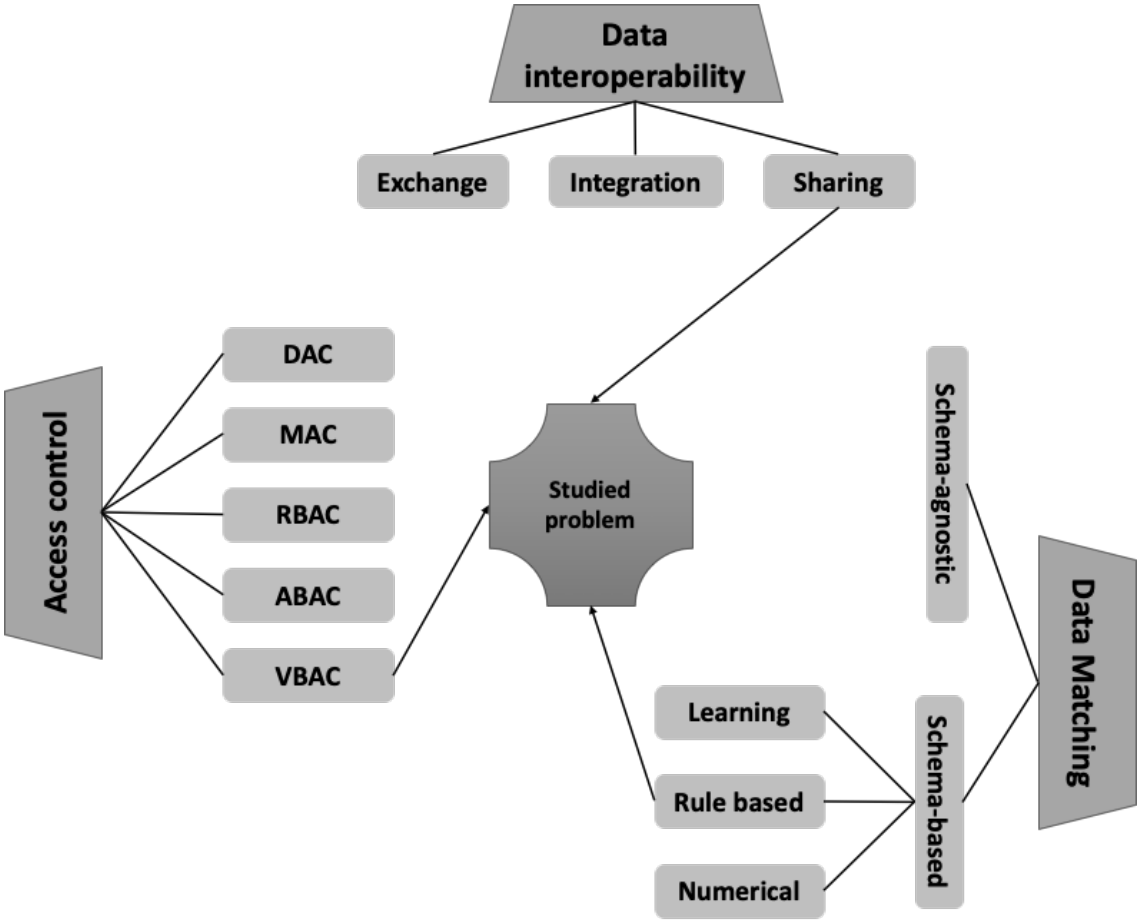


Figure 5.1: Studied problem with respect to related work

particular instance that we believe is the most relevant. Next, we discuss details of the problem with respect to each dimension.

5.3.1 | Data interoperability

Sharing data allows the data management systems to exchange and consume data based on clear and accurate expectations; the content (i.e., without changing any local setting: storage, schema, policy, and infrastructure). Indeed, we aim at considering a data sharing setting where it is possible to consume data from several sources without moving the latter. Therefore, we choose the data sharing configuration among existing models (e.g., data integration, exchange ...) for its ability to handle heterogeneity and restrictiveness occurring from peer schemas. This means that the sharing is using a mapping at data-level.

Data integration and data exchange models rely mainly on the schema to build the querying mechanism. However, if any of the data sources modifies its schema, the overall system will be disturbed. In contrast, as the mapping in data sharing are instance driven, local schema modification will not impact the global sharing mechanism. In addition, if we assume that the content of local sources are constantly updated then the data sharing is a powerful mechanism to handle this situation where one has does not to synchronize with other peers.

5.3.2 | Access control Model

We adopted view-based access control model to enforce security between the participating data sources. A view offers a better representation of data that needs access restriction. Since we are in the optic of sharing data we consider the hypothesis that data owners are motivated to maximize the sharing rather than limiting access. As the behaviour of local sources could change access control must fit the requirements in such a dynamic environment. For each role or user (depending on the local configuration) a list of policy rules is attached. When a user is logged into the system each of her/his queries are executed under the attached policy rules. The choice to use VBAC is because of its ability to reach finer granularity and the flexibility to simulate other models of classical approaches (DAC,MAC and RBAC).

An important key we consider in this thesis is the inference that could occur from overlapping data. Policy views limit access to specific data through attribute association. Nevertheless, the mechanism of access control alone in such an environment shows its limits, specifically due to the duplication of related information present in different sources.

5.3.3 | Data matching model

Data matching is, traditionally, used to match pairs of records referring to the same real-world object. We considered schema-based methods since we apply our approach on relational data model. Our choice turned into the use of a rule-based approach because of its interpretability for humans which is supported by other systems such as ML methods. The rule-based approach offers to our configuration the possibility to describe the matching conditions. Thus, rules are used as mappings between data sources to link the instances.

Entity matching rules are established in hand-crafted expert who have sufficient knowledge about data. The rules are then involved in an online process for querying in the overall data sharing system. Rule-based approaches are very interesting as it is possible to improve the quality rules, for better effectiveness of the entity matching process, through an hybridization between human knowledge and program synthesis [Sin+17].

5.4 | Overall problem statement

Given two data sources, each one having its own access control policy, and entity matching rules between both sources:

- *Are the views used to define data willing to be shared violating local security policies? How could we revise views that violate access control rules to improve data availability?*
- *How to manage the querying in data sharing setting with several parties in total accordance with their (and other) local security policies?*

5.5 | Methodology

The proposed approach, as shown in Figure 5.2 addresses the problem on two complementary levels: (1) Prepare data source for publishing data without violating security policies, (2) Enforce access control at the global sharing framework. The methodology consists in addressing the problem in two stages:

- Offline stage: It aims at assisting administrator to check that the data to be shared are not violating the security policy. This is achieved at design time by comparing views to the access control policy rules.

  Our method for detecting and revising privacy violations in this context tackles the data publishing problem. It defines a published view V (containing data to be shared later) and a security policy S to preserve the privacy. When V and S return no tuples in common, over all possible instances this means that V preserves S. Indeed, we establish necessary and sufficient conditions based on the concept of *disjoint queries* to characterize when a view preserves security policies. Two tasks are applied in this stage: (i) Identifying views

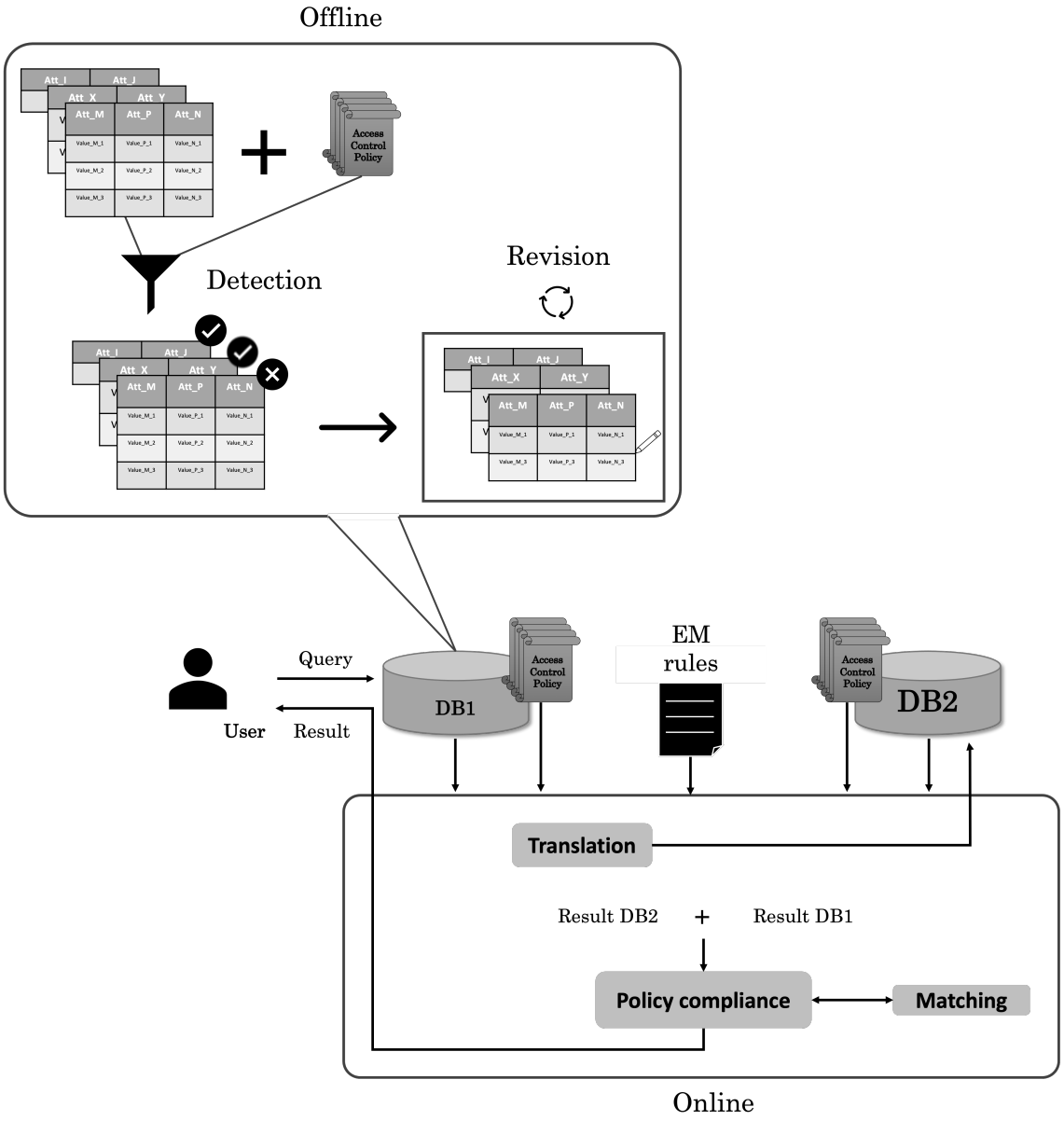


Figure 5.2: Methodology

violating access control rules, i.e., presenting high risk for disclosing sensitive information. (ii) Propose a revision for views that violate the access control rules to ensure more data availability.

- Online stage: This stage is mainly applied based on a trusted third party. It takes as input the local access control rules, the entity matching rules, and a submitted query. The goal is to extend answers of the local sources with possible answer from the other sources without disclosing any sensitive information of those data sources. In particular, it tracks overlapping information of one real-world entity residing in the different data sources. It includes the three following phases:
  - Translation: This phase translates the local query into an equivalent query to be submitted to the other data source based on the entity matching rules.
  - Matching: It consists in identifying all the matching and non-matching records from both results of both sources. This output of this phase is used in the next phase.
  - Policy compliance: It transforms the sets of returned records from both data sources into a final result that complies with security policies. It eliminates from the set of results the records that violate access control rules.

Next chapters elaborate on these levels. Chapter 4 addresses the data publishing problem from the perspective of the compliance of views with local access security policies. Chapter 5 discusses the data sharing framework with two solutions based on a trusted third party: (i) a naïve approach where all local access control policies are shared, and (ii) an approach that keeps local access control hidden to avoid understanding of the policy restriction mechanism.

Part II

Contributions

6

Data Publishing

Contents

6.1 | Introduction

With the rapid growth of stored information, governments and companies are often motivated to derive valuable information. Enterprises and organizations have begun to employ advanced techniques to analyze the data and extract "*useful*" information. Data publishing has fueled significant interest in the database community [Cli+04]. It is a widespread solution to make the data available publicly and enable data sharing. It consists on exporting or publishing information of an underlying database through views to be used by peers. At the same time, the publisher tries to ensure that sensitive information will not leak.

There are great opportunities associated with data publishing, but also associated risks [Sha+19]. Indeed, it is complex to find a trade-off between preventing the inappropriate disclosure of sensitive information and guarantee the availability of non-sensitive data: Removing of all the data exported by a violating view achieves perfect privacy, but it is a total uselessness, while publishing the entire data, is at the other extreme.

In this chapter, we focus on a variant problem of data availability referred also as utility of the data [RSH07], and we specifically consider data privacy in database publishing. The owner of a given database D wishes to publish a set of views $\mathcal{V}$ under a set of restrictions $\mathcal{S}$. We assume that the restrictions on the views are expressed as a set of queries, called policy queries [ND07; MS07]. Inspired by prior work on privacy-preservation [VMI09], we define a view to be *safe* w.r.t. a policy query if they don't return tuples in common, in other words, the view and the policy query are disjoint. We provide a revision algorithm for the privacy-preservation protocol to ensure data availability [AH20]. Indeed, instead of neutralizing the unsafe view, we propose a rewriting technique that exploits the interaction between a view and a policy query to enrich the view with relevant information for ensuring the availability of data. Our approach exploits the concept of residue introduced in [CGM90].

6.2 | Motivation scenario

We illustrate our data publishing setting by the following running example. Let D be a database schema of a company consisting of two relations:

$$Employee(id\_emp, name, dept)$$
$$PayrollService(id\_emp\_ps, accountNum, salary)$$

The relation *Employee* stores, for each employee her/his identifier, her/his name and her/his department. The relation *PayrollService* stores information related to employees, in particular, their identifier, their account number, and their salary.

Consider the following set $\mathcal{S}$ of policy queries. The policy queries define the

information that is sensitive to make secret to public.

$$S_1(i,n) : -Employee(i,n,d)$$
$$S_2(n,s) : -Employee(i,n,d),\ PayrollService(i,a,s),\ s > 3000$$

Query S_1 projects the *identifier* and the *name* of an employee. It states that both attributes are sensitive when they are returned simultaneously. S_2 states that the *name* and the *salary* of employees with a salary over $3 000 are sensitive if returned together.

Now, consider the following set of published views $\mathcal{V}$. The view V_1 projects the *name* and the *salary* of the employees in department "*info501*" whereas V_2 projects the *name* and the *salary* of the employees with a salary under $10 000.

$$V_1(n,s) : -Employee(i,n,d),\ PayrollService(i,a,s),\ d =' info501'$$
$$V_2(n,s) : -Employee(i,n,d),\ PayrollService(i,a,s),\ s < 10000$$

The policy query S_1 is not violated by any of the views in $\mathcal{V}$ since none of them returns the *identifier* and the *name* of the employees. The view V_1 discloses some sensitive information since $S_2(I)$ and $V_1(I)$ overlap for some database instances, such as I= *{Employee { ⟨'p552','Jhon','info501' ⟩}, PayrollService { ⟨'p552','FRB1015', 4500 ⟩ }}*. Regarding the view V_2 and the query S_2, one can notice that there could be some overlapping tuples since the selection conditions in the two queries are both satisfiable for some values of s.

6.3 | Preliminaries

In this section, we introduce the relevant concepts to our framework. We consider a relational setting. A *database schema* D consists of a finite set of relation schemas $R_1, ..., R_n$, where each relational schema (or just *relation*) consists of a unique name and a finite set of attributes. The set of attributes in a relational schema R_i is denoted by $att(R_i)$. A *tuple* for a relational schema R_i, where $att(R_i) = \{X_1, ..., X_k\}$ is an element consisting of a set of k constants. A relation r defined over a relational schema R_i, denoted by $r(R_i)$ (*or simply by* r_i *if* R_i *is understood*) consists

of a finite set of tuples defined over R_i. A database instance defined over a schema $D = R_1, ..., R_n$, denoted by $I(D)$(*or simply by I if D is understood*) is a finite set of relations $\{r_1(R_1), ..., r_n(R_n)\}$. We consider nonrecursive DATALOG queries with inequalities defined as follows. For the simplicity of the presentation, we assume that Boolean queries are not allowed[1]. Hence, the head of the query contains only variables and at least one variable. Despite the fact that constants do not appear in the head of the query, the essentials of our results could be extended for this case.

6.3.1 | Definitions

Definition 1. *We assume a set of variable names $\mathcal{N}$, and the function typ: $\mathcal{N} \rightarrow att(R_1) \cup ... \cup att(R_n)$, where $D = \{R_1, ..., R_n\}$. A conjunctive query Q is defined by:*

$$Q(\bar{X}) : -t_1, ..., t_n, C_{n+1}, ..., C_{n+m}$$

where $t_1, ..., t_n$ are terms, $C_{n+1}, ..., C_{n+m}$ are inequalities, and $\bar{X} \in \mathcal{N}$. A term is of the form $R(X_1, ..., X_n)$, where R is a relational schema in D and $\{X_1, ..., X_n\} \subseteq \mathcal{N}$. An inequality has one of the following form:

1. *$X \odot c$, where $X \in \mathcal{N}$, c is a constant (i.e., numeric or string), $\odot \in \{=, \leq, \geq, <, >, \neq\}$.*
2. *$X \odot Y$, where $\{X, Y\} \subseteq \mathcal{N}$, $\odot \in \{=, \leq, \geq, <, >, \neq\}$.*

$Q(\bar{X})$ is called the head of the query Q, and $t_1, ..., t_n, C_{n+1}, ..., C_{n+m}$ is called the body of Q. $\bar{X}$ is called the schema of Q and is also denoted by $att(Q)$. The queries have the following restrictions:

a. *We assume that a variable can appear at most once in the head of a query.*
b. *If a variable X_i appears as the i^{th} variable in a term $R(, ..., Xi, ...)$ then $typ(X_i) = A_i$, where A_i is the i^{th} attribute of R;*
c. *$\bar{X} \neq \emptyset$.*

[1]Policy queries specify tuples to keep private. Thus, a policy query with a set of variables empty in the head is useless.

We recall that the views are considered to be conjunctive queries in our framework. Condition (a) above does not prevent join and self-join queries from being defined in our framework. Condition (c) mentions clearly that Boolean queries are not allowed. We also assume that the queries are range restricted (safety of a DATALOG query), *i.e.,* every variable X in $\bar{X}$ also appears in some term in the body of Q, or there exists a variable Y such that $C_{n+1}, ..., C_{n+m}$ imply that $X = Y$ and Y appears in some term in the body of Q.

Next we will introduce some notations needed to compare the schema of queries based on the types of the variables.

Definition 2. *Given queries $S(X_1, ..., X_n)$ and $V(X'_1, ..., X'_m)$, $att(S)$ is contained in $att(V)$, denoted by $att(S) \preceq att(V)$, if $n \leq m$ and $typ(X_i) = typ(X'_i)$ for all $i \in \{1, ..., n\}$. The schemas $att(S)$ and $att(V)$ are equivalent, denoted by $att(S) \equiv att(V)$, if $att(S) \preceq att(V)$ and $att(V) \preceq att(S)$. The difference between $att(V)$ and $att(S)$, denoted by $\diagdown_{typ}$ is defined as:*

$$att(V) \diagdown_{typ} att(S) = \{X'_i | \nexists X_j (typ(X'_i) = typ(X_j))\}.$$

6.3.2 | Problem definition

The questions addressed are the following : **Are the published views $\mathcal{V}$ safe *w.r.t.* the policy queries $\mathcal{S}$? If the views are not safe *w.r.t.* the policy queries, how could we revise them to make available the subset of tuples which are not in the set of tuples identified by $\mathcal{S}$?**

6.4 | Privacy preservation

Our notion of privacy preservation is build upon the protocol introduced in [VMI09]. Below, we formalize the notion of privacy preservation as defined in [VMI09] and in the next section we extend it for the query revision. First, we summarize the notion of disjoint queries.

Definition 3. *If I is an instance of a database schema D and S and V are queries that have the same schema, then S and V are defined to be disjoint if for every I in*

D, $S(I) \cap V(I) = \emptyset$.

The adopted approach for defining privacy consists in specifying the information that is private by a query S, and then the notion of a user query being legal translates to the requirement of our approach that the user query must be disjoint from S for all possible database states. The privacy violation occurs only when the same tuple is returned by both the policy query and the user query, for some database instance. The basic semantic unit of information is a tuple, not an attribute value therefore the intersection is at the tuple level and not the attribute level.

The privacy preservation we want to achieve can be determined using only the structure of S and V, i.e., independently from data which can be checked at compile time. We define privacy preservation as follows.

Definition 4. *Query V is privacy-preserving with respect to a policy query S if either:*

a. $att(S) \diagdown\!\!\!\!\diagdown_{typ} att(V) \neq \emptyset$*;or*

b. $att(S) \preceq att(V)$ *and S and V' are disjoint, where V' is the query defined by:* $V'(att(S)) : -V$

Essentially, the case (a) is the situation where there are some attributes of S that are not in V, for which we do not consider to be a violation. It could be illustrated by V_1 and S_1 from our running example. As it is previously discussed, we do not consider V_1 to be a privacy violation for S_1. In the situation of case (b) every variable in $att(S)$ has a matching variable in $att(V)$, in which case we require that the projection of V on $att(S)$ be disjoint from S.

The main result established in [VMI09] characterizes when a published query V preserves the privacy of a secret query S. So, given S and V two DATALOG queries over a database scheme D such that $\bar{C_S}$ and $\bar{C_V}$ are separately satisfiable, then V preserves the privacy of S if either:

1. $att(S) \diagdown\!\!\!\!\diagdown_{typ} att(V) \neq \emptyset$; or
2. the set of inequalities $\pi_{att(S)}(\bar{C_S}) \cup \pi_{att(S)}(\bar{C_V})$ is unsatisfiable.

Next, we illustrate by an example the detection of privacy violations of a published view *w.r.t.* a policy query.

Example 7. *We demonstrate over a simplified version of the running example, where the set of policy queries* $\mathcal{S} = \{S_2\}$ *and the set of published views* $\mathcal{V} = \{V_3\}$.

$$V_3(n, a, s) : -Employee(i, n, d),\ PayrollService(i, a, s),\ s \leq 5000,\ d =' info501'$$

We note that $\bar{C}_{V_3} = \{s \leq 5000,\ d =' info501'\}$ *and* $\bar{C}_{S_2} = \{s > 3000\}$. *Then* $\pi_{att(S_2)}(\bar{C}_{S_2}) = s > 3000$ *and* $\pi_{att(S_2)}(\bar{C}_{V_3}) = \{s \leq 5000\}$. *We say that* V_3 *is not privacy-preserving w.r.t. policy query* S_2 *since* $att(S_2) \underset{typ}{\setminus} att(V_3) = \emptyset$ *and* $\pi_{att(S_2)}(\bar{C}_{S_2}) \cup \pi_{att(S)}(\bar{C_V}) = \{s > 3000, s \leq 5000\}$ *which is satisfiable.*

6.5 | Revising privacy violating views

In the previous section we introduced the privacy preservation and described how to detect the violating views. In this Section we present a technique for revising the views w.r.t. policy queries, over all database instances.

Our approach to revise the views is based on the explorations of residues resulting after the unification process. This technique is used for semantic query optimization [CGM90]. The unification process allows to capture residues from policy queries and will be associated with published views.

Revising the views can be described informally as the process of transforming the non-privacy-preserving view to comply with the requirements of the policy query. When the schema of the view and the query is the same (*see Definition 4*), this means that bodies are unsatisfiable. In this case, we propose using the partial subsumption technique, to extract the residue of a policy query and associate it with the view. A policy query S partially subsumes a view V if a subclause [2] of S subsumes the body of V.

In the following, the process is going to be illustrated by an example. Consider a policy query S and a view to be published V over a database with two relations $R_1(a, b, c)$ and $R_2(a, c)$. The policy query S states that the attribute association $\{a, b\}$ is sensitive for tuples in R_1 that could be join with R_2 on the attribute a and where c is greater than 5. The view V project the attribute a' and b' for tuples in

[2] C is a subclause of D if every literal in C is also in D. A literal could be either a term or an inequality

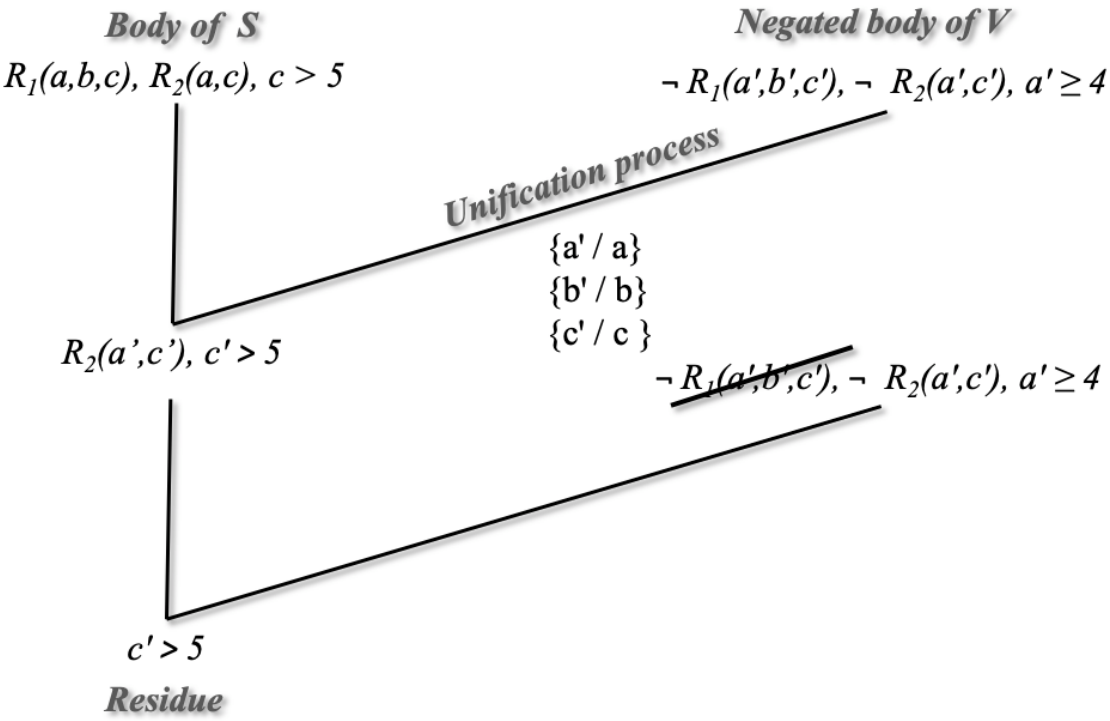


Figure 6.1: Refutation tree

R_1 join R_2 on *a'* and where *a'* is under 4.

$$S(a,b) : -R_1(a,b,c), R_2(a,c), c > 5$$
$$V(a',b') : -R_1(a',b',c'), R_2(a',c'), a' < 4$$

Since V is not privacy-preserving to S we proceed to the revision the view. It consists on generating the residue from the policy query S using the V using the partial subsumption technique. First, the body of V is negated and, thereby, the set $\{\neg R_1(a',b',c'), \neg R_2(a',c'), a' \geq 4\}$ is obtained.

Then, we construct a linear refutation tree (*see Fig. 6.1*) by considering the body of S as the root. At each step, an element of the negated body of V is unified using an element in the root: The first step of the refutation tree, unify $\neg R_1(a',b',c')$ and $R_1(a,b,c)$ with the following substitutions $\{a'/a, b'/b, c'/c\}$. In the next step, $\neg R_2(a',c')$ of the negated body V is unified with $R_2(a',c')$ in S according to the substitutions of the previous step.

Finally, we obtain at the bottom of the tree the residue: $c' > 5$, which could

interpreted as "c' *cannot be over 5*". The residue is then associated to the view V :

$$V^r(a', b') : -R_1(a', b', c'), R_2(a', c'), a' < 4, \{c' > 5\}$$

According to the interpretation before, the meaning of the previous view is as follows:

$$V^r(a', b') : -R_1(a', b', c'), R_2(a', c'), a' < 4, c' \leq 5$$

Revising views depends on the type of resulting residue. For instance, if an empty set is obtained at the bottom of the refutation tree called *null residue*, we would associate to the view an empty residue { } which is then replaced by a $false$ term. A false term in a query leads to directly exclude the view from the set of published views $\mathcal{V}$. We recall that we omit the case of unsatisfied residue since we consider only the non-privacy-preserving views for revision.

Above we assumed V and S having the same schema. Now, we consider the case where some elements of the schema $att(S)$ appears in $att(V)$(*i.e.,* $att(S) \underset{typ}{\diagdown} att(V) = \emptyset$ and $att(S) \preceq att(V)$). Same as previously, the residue would be computed using the body of the view V and S and added to the given view. In case the result of the refutation tree is empty, instead of incorporating the term $false$, which implies removing the entire view as seen before, we propose to keep the view and put the sensitive attributes as secret. Indeed, the attributes of $att(S)$ in the view V are replaced by $NULL$. Putting the sensitive attributes to $NULL$ prevents the disclosure and helps to redesign the views for a better utility.

Example 8. *Consider the view V_1 from our running example and the following policy query S_3:*

$$S_3(n) : -Employee(i, n, d), PayrollService(i, a, s)$$

The result of the refutation tree would be a null residue and we note that $att(S_3) \underset{typ}{\diagdown} att(V_1) = \emptyset$ and $att(S_3) \preceq att(V_1)$. In this case, we propose to rewrite V_1 into V^r having the following form:

$$V^r{}_1(NULL, s) : -Employee(i, n, d), PayrollService(i, a, s), d =' info501'.$$

6.6 | Summary

We addressed the problem of privacy-preserving and data availability in data publishing under policy queries. We proposed a preliminary method to revise views. The method is based on the concept of residue usually adopted in semantic query optimization. Our work aims to strike the balance between data privacy and data availability. Indeed, instead of only neutralizing the non-privacy-preserving views, we proposed a data-independent process for revising the views *w.r.t.* policy queries that sanitize the views from sensitive information.

Our revising model returns changed views that do not guarantee the correctness since some information could be hidden. However, it provides safe access to the published data and considerable availability.

7

Data Sharing

Contents

7.1 | Introduction

Widespread use of data in many different sectors empowers the quality of decision-making. Indeed, large volumes of information are shared between several sources for data analysis purposes [LJ12; Wu+14]. Data sharing is one of the configurations that allows sources hosting data sets complying with specific schemas to share information. To allow data sharing between the acquainted sources, mappings are necessary and take the form of data-level and schema-level.

In this chapter, we consider the data sharing problem which differs from the previous problems since it is based on the notion of mapping that encompasses both schema-level (*attribute-to-attribute*) and data-level (*value-to-value*) mappings [KA04; KAM03b; Elm+10].

Data in different and heterogeneous sources may overlap or be closely associated. Entity matching is used to identify the same real-world object from different sources even if it is represented differently (different formats, spellings etc.) or containing errors. The data heterogeneity problem arises when the same real-world object is represented using different identifiers in different sources. For example, a patient may be uniquely identified using the social security number (*SSN*) in an hospital and by a *donor_id* in a blood bank service.

Sharing data between different sources enriches the context in which the data are used. Nonetheless, it could potentially reveal sensitive information. In several areas, information that sets protected against unwarranted disclosure is considered as sensitive. Indeed, data owners enforce their own security policy to control access to contents. Access controls may also differ from one source to another as they are independent and autonomous. However, some of the information shared could be sensitive in a given source and not sensitive in another one. For some pieces of data that may be closely associated and similar[1], access could be denied in one source, whereas it may be granted in another source. Hence, one faces a violation of a security policy.

There are several motivating scenarios for data sharing meeting with privacy in numerous real-world applications. For example, sharing healthcare data could improve scientific research. It can, for instance, enable early detection of disease outbreak [Cli+04]. However, obtaining consent to use information can be risky.

[1]Semantically equivalent, they denote the same real-world object

Indeed, disclosure of sensitive personal information can lead to a serious damage for the concerned individuals.

In this chapter, we focus data sharing framework in the presence of access control policies associated with the involved sources. Data sharing is materialized by entity matching rules considered at data-level rather than at schema-level. Our current work describes a methodology that includes a set of transformation and verification steps to restrict answer to queries over a data sharing system, where data owners have complete control of their own data. Our main contributions are:

- We introduce formal definition of query answering in data sharing based on entity matching that enforces access control. We present a technique for translating queries using entity matching rules.
- We introduce the rewriting of queries to enforce access control policies of collaborating sources.
- We present two detailed strategies to achieve query answering in data sharing complying with local access control policies based on entity matching [**agoun2021data**].

The problem we study was described as an open research challenge in [Cli+04]. To the best of our knowledge, our work is the first to provide a practical methodology for view-based access control in a data sharing paradigm, taking into account mappings between entities from different participating data sources.

7.2 | Motivating Scenario

Let us consider two institutions, a *hospital* and a *blood bank* that agree on sharing some subsets of their data sets. The *hospital* stores data about patients in a relation we denote *patient*(*SSN, name_p, address_p, city_p, sex, blood_pressure, blood_glucose, height_weight, diagnosis*) where *SSN* denotes the social security number, *name_p* denotes the name, *address_p* denotes the address, *city_p* denotes city, *sex* denotes the gender,
blood_pressure is the measured blood pressure,
blood_glucose is the blood glucose level, *height_weight* is the height and weight, and *diagnosis* denotes the pathology of the patient.

Table 7.1: Instance of *patient*

SSN	name_p	address_p	city_p	sex	blood_pressure	blood_glucose	height_weight	diagnosis
738-77-8987	Bob Tracy	3 rue emile zola	lyonn	M	120/79	73	162/71	headache
358-87-9526	Smith, John	06 bis rue notre dame	paris 6	M	126/76	71	182/85	stomachache
852-37-9526	Tim McCall	43 av. des Postes	Lille center	M	124/75	131	175/42	diabetes
436-44-0945	Jeane Henri	48, rue du Four	75006 Paris	F	146/97	69	156/52	Hypertension

Table 7.2: Instance of *donor*

id_d	donor_name	donor_address	donor_city	gender	blood_pressure	blood_glucose	h_w	number_donation
455	Robert Tracy	03 rue emile Zola	lyon	Male	120/79	71	162/72	2
589	John A. Smith	06 bis rue notre dame	paris	Male	127/77	70	181/89	4
996	Timothy McCall	43 aveunue des Postes	lille	Male	121/73	136	176/53	0
195	Marine.P Jolio	48 B.v pierre marion	marseille	Female	116/77	72	163/55	1

The *blood bank institution* stores, in the relation *donor*(*id_d, donor _name, donor_address, donor_city, gender, blood_pressure, blood_glucose, h_w, number_donation*), information related to donors. The attribute *id_d* stands for the donor's id, *donor_name* is the donor's name, *donor_address* corresponds to the address, *donor_city* stands for the city, *gender* indicates the donor's gender, *blood_pressure* refers to a measured blood pressure, *blood_glucose* refers to the blood glucose level, *h_w* denotes the donor's height and weight, *number_donation* represents the accumulated number of blood donations made by a donor.

The two relations *patient* and *donor* agree on an *attribute alignment* that maps the *patient* attributes *name_p, address_p, city_p, sex, blood_pressure, blood_glucose, height_weight* to *donor_name, donor_address, donor_city, gender, blood_pressure, blood_glucose, h_w* of the relation *donor*, respectively.

The two databases display entity heterogeneity as they do not share a common identifier (*see tables 7.1 and 7.2*). Some information describing a donor might be associated with a given patient in the hospital database if both tuples refer to the same real-world individual. Indeed, a donor could have dealt with the hospital for a particular reason. For instance, in the given blood bank database instance, the name can be spelled as `John A. Smith`, while in the hospital database, it can be spelled as `Smith, John`. The two entities, patient and donor in this case, represent the same real-world person with similarities in name, address, city and sex. The similarities are computed according to an entity matching rule denoted ϕ_{EM} (*more details will be provided later on*). Given two tuples $p \in$ *patient* and $d \in$ *donor*, the entity matching rule in our scenario is expressed as follows:

$$\begin{aligned}\Phi_{EM} = \quad & p[name\_p] \approx_{(Jaro,78)} d[donor\_name] \wedge \\ & p[address\_p] \approx_{(Levenshtein,72)} d[donor\_address] \wedge \\ & p[city\_p] \approx_{(Smith-Waterman,77)} d[donor\_city] \wedge \\ & p[sex] \approx_{(jaro,70)} d[gender].\end{aligned}$$

The entity matching rule is computed over a subset of the aligned attributes where $\approx_{(f,\epsilon)}$ is the corresponding similarity function f and ϵ the threshold (*see definition 5*). Two records p and d *match* iff Φ_{EM} is evaluated to *True*.

Please note that there are some records that are similar since they satisfy Φ_{EM}, *see Tables 7.1 and 7.2*. For instance, the record *Tim McCall* with the *SSN* "852-37-9526", in the *patient* instance, has a similar record with the *id_d* "996" in the *donor* instance, as it satisfies Φ_{EM} with the following evaluation:

$$\begin{gathered}Jaro(\text{ Tim McCall, Timothy McCall}) = 90 \wedge \\ Levenshtein(\text{ 43 av. des Postes, 43 aveunue des Postes}) = 76 \wedge \\ Smith - Waterman(\text{Lille center, lille}) = 80 \wedge \\ jaro(M, Male) = 75.\end{gathered}$$

In our setting, we consider the rules expressed as *forbidden views* [KR11], also called *secret views* (*see definition 9*). Each data source sets forth a security policy expressed as a set of access control rules.

The hospital database denies access simultaneously to both attributes *SSN* and *diagnosis* (*expressed in rule* r_1), while also denying access simultaneously to both attributes *name* and *diagnosis* (*rule* r_2) for all the patients. The rules r_1 and r_2 express the access control policy $\Pi_{patient}$ associated with the relation *patient*:

r_1 : **Deny** SSN, diagnosis
From patient

r_2 : **Deny** name_p, diagnosis
From patient

The blood bank database, on the other hand, denies access simultaneously to the combination of name and blood pressure (*rule* r_1'), and the association of the donor's id and blood pressure (*rule* r_2') for all the donors living in *Lille*. The access control policy Π_{donor} associated with the relation *donor* is the set of the two rules:

r'_1 : **Deny** donor_name ,
blood_glucose
From donor
Where donor_city = 'lille '

r'_2 : **Deny** id_d, blood_glucose
From donor
Where donor_city = 'lille'

Now, assume we want to retrieve all the male patients with their *name, city, blood_pressure, blood_glucose* and their *height_weigh* from *patient*. Such a query can be expressed as:

q : **Select** name_p, city_p,
blood_pressure,
blood_glucose,
height_weight
From patient
Where sex = 'M'

The attribute alignment and the entity matching rule between *patient* and *donor* might provide sufficient information to translate the query q over *patient* in the hospital database to a query q' over the *donor* relation in the blood bank database. The derived query q' could be of the following form (*details will be given in the next section*):

q' : **Select** donor_name, donor_city,
blood_pressure,
blood_glucose, h_w
From donor
Where gender = 'Male'

The tuples returned from the evaluation of q are displayed in *Table 7.3*.

The results of the query q' are shown in *Table 7.4*. Please note that the tuple with the name **Timothy McCall** living in Lille is not returned because it is denied by the rule r'_1.

Now, we show the access control violation that could occur from the returned query if no mechanism for secure data sharing is put in place. If we look at the results of *Tables 7.3* and *7.4*, and based on the entity matching rule Φ_{EM}, the tuple

Table 7.3: Retrieved records from q

name_p	city_p	blood_pressure	blood_glucose	height_weight
Bob Tracy	lyonn	120/79	73	162/71
Smith, John	paris 6	126/76	71	182/85
Tim McCall	Lille center	124/75	131	175/42

Table 7.4: Retrieved records from q'

donor_name	donor_city	gender	blood_pressure	blood_glucose	h_w
Robert Tracy	lyon	Male	120/79	71	162/72
John A. Smith	paris	Male	127/77	70	181/89

corresponding to the record *Tim McCall* is returned by the evaluation of q, while the tuple it matches in the *donor* relation namely with *Timothy McCall'*. The latter was not returned in the answer of q' because the access to the association of name and blood glucose for a donor living in Lille is denied.

Thus, in a data sharing setting, retrieving information from one side, the hospital database (here *patient* relation), leads to a violation of the blood bank access control policy. This is a disclosure of a sensitive information. This example highlights a violation of an access control policy in the context of data sharing.

7.3 | Preliminaries

We consider SQL queries and views without grouping and aggregation. For a notational convenience, we modify the naming convention of standard SQL to guarantee unique attribute names for each of the attributes in a query. Let $R(A_1, A_2, ..., A_n)$ be a relation with n attributes denoted $att(R)$. Let t be a tuple in R. The notation $t[A_i]$ stands for the value of the attribute A_i in the tuple t. Given a query q, we use $Tables(q)$ to denote the set of tables in the FROM clause. The set of attributes in the SELECT clause is denoted by $Sel(q)$. The conditions in the WHERE clause are denoted $Conds(q)$ a Boolean combination of constraints (*see definition 10*).

In the following section, we introduce important notions needed in our methodology. Then, we formally define the problem of data sharing under access control policies.

7.3.1 | Entity matching

Entity matching definitions are mostly based on [Sin+17].

Definition 5. *(Similarity function) Let v and v' be two values from the same domain. The similarity function f computes a positive real score in the interval $[0, 1]$ corresponding to the distance between the values v and v' :*

$$f(v, v') \in [0, 1]$$

A higher score means that v and v' have a high similarity.

Definition 6. *(Attribute Matching Rule) Given two relations $R(A_1, ..., A_n)$ and $R'(A'_1, ..., A'_n)$, two tuples t and t' in R and R', respectively, a similarity function f, and a positive real number θ called threshold. An attribute matching rule is a Boolean function $f(t[A_i], t'[A'_i]) \geq \theta$. Evaluating an attribute matching rule to True means that the attribute value $t[A_i]$ matches $t'[A'_i]$, and we write $t[A_i] \approx_{(f,\theta)} t'[A'_i]$ as the attribute matching rule for the similarity function f and the threshold θ.*

Definition 7. *(Entity Matching Rule) A matching rule Φ between two relations R and R' is a conjunction of attribute matching rules of the form:*

$$\Phi = t[A_1] \approx_1 t'[A_1] \wedge ... \wedge t[A_n] \approx_n t'[A_n].$$

Roughly speaking, two tuples $t \in R$ and $t' \in R'$ are similar ($t \sim_\Phi t'$) and we say they match iff Φ is evaluated to True.

Definition 8. *(Attribute alignment) An attribute alignment from R to R' is a 1-1 mapping denoted $\boldsymbol{m}$ from $M_{att(R)}$ to $M_{att(R')}$, where $M_{att(R)} \subseteq att(R)$ and $M_{att(R')} \subseteq att(R')$, such that if A is an attribute in $M_{att(R')}$, then, there exists an attribute A' in $M_{att(R')}$, such that $A' = m(A)$.*

In the rest of the paper, we denote by $M^{\Phi}_{att(R)}$ (*resp.* $M^{\Phi}_{att(R')}$) the set of attributes in $att(R)$ (*resp.* $att(R')$) that appear in Φ.

7.3.2 | Access Control

Access control aims at restricting the actions or operations that a legitimate user can perform in a system (e.g., operating system, DBMS). To achieve those goals,

different concepts have been defined. The security policy describes the system requirements for complying with some specification (e.g., laws, regulations). The security model formalizes the rules defined in the security policy and describes how they should work. The security mechanism describes the low level methods used to enforce the formalized rules.

Definition 9. *(Access control rule) An access control rule is a view that specifies the part of data for which access is denied. The syntax of an access control rule is identical to that of a select-project-join (SPJ) query without any* ***distinct*** *in the* ***Select*** *attribute list. As an exception,* ***Deny*** *replaces the keyword* ***Select****.*

Deny	attribute list
From	relation list
Where	condition list

The ***Deny*** *clause specifies the prohibited combination of attributes for the specified relation in the* ***From*** *clause. The* ***Where*** *clause states which constraints a given tuple should satisfy in order not to be disclosed.*

Tuples for which access is denied by one or more access control rules are considered as sensitive.

Definition 10. *(Constraint) A constraint is a comparison predicate of the form* $x\mathrm{Y}y$*, where* x *and* y *are terms formed from attributes or constants (but not both constants) and* Y *is a comparison operator taken from the set* $\{=, \leq, \geq, <, >, \neq\}$.

Please note that to simplify the notation, $Sel(r)$ refers to the set of attributes in the Deny clause of r, where r stands for an access control rule.

7.3.3 | Problem formulation

Given two databases D and D' where $R \in D$ and $R' \in D'$ two relations without duplicates, an attribute alignment m between R and R', and an entity matching rule Φ.

Each database provides its own security policy Π_R and $\Pi_{R'}$, attached to R and R', respectively. We are concerned with providing a framework where several parties could share their data in a complete accordance with their local security policies and the entity matching rule. In other words, given a query q over R and

for an equivalent translated query q' over R', if q could access and retrieve a given tuple $t \in R$ and q' could retrieve $t' \in R'$ such that $(t \sim_{\Phi} t')$ but access is denied in D', the restriction should be applied to q in D.

7.4 | Query translation

In the following, we discuss how to derive a query q' from a query q given an entity matching rule Φ and the attribute alignment m between the relations R and R'.

We assume the sources to be autonomous and to have different schemas. The query q may involve some attributes that could not be mapped to another source. To make the translation possible, some conditions need to be met before translating a query.

Translation conditions. For a correct translation of q over R into q' over R', the query q must satisfy the following conditions:

1. All the selection attributes $Sel(q)$ must appear in the set of aligned attributes $M_{att(R)}$.
2. All the attributes involved in $Conds(q)$ must appear in the aligned attributes $M_{att(R)}$.

To translate a query, given an entity matching rule and an attribute alignment, we proceed in two steps.

Step 1. We consider a query q that satisfies the translation conditions. Each attribute in q is substituted by its mapping attribute in R' based on the attribute alignment m. The obtained query q' has the following properties:

1. $Sel(q') = \{a' \mid a' = m(a) \text{ and } a \in Sel(q)\}$.
2. $Tables(q')$ is the target relation of $Tables(q)$.
3. $Conds(q')$ is the set of constraints obtained by simultaneously substituting each attribute in $Conds(q)$ by the corresponding attribute in $M_{att(R')}$ following m.

Example 9. *Let us consider the query q of the motivating example (Section 7.2). Since all the attributes in $Sel(q)$ could be mapped to their corresponding attributes in the relation donor, the following query is obtained:*

q' :	**Select**	donor_name,	donor_city,
		blood_pressure,	blood_glucose,
		h_w	
	From	donor	
	Where	gender = 'M'	

The first step is mainly a substitution phase, where attributes in q are replaced by their corresponding attributes in R'. However, in some cases, the constant values used in the WHERE clause may differ from one source to another.

Step 2. Constants may differ from one data source to an other. Hence, in this step we focus on constants in $Conds(q')$. As an example, in the relation *donor* the gender can be either $Male$ or $Female$, whereas in the relation *patient* it is expressed as M or F. Similarity predicates are used as constraints to approximate the constants. Let us first introduce the definition of *similarity predicate* [OCM02].

Definition 11. *(Similarity predicate) A similarity predicate is a function with three parameters: (i) a set of attribute values (ii) a value to be compared (iii) a threshold value k in the range $[0, 1]$:*

$$Similarity\_Predicate\_Function(\text{ set of attribute values, input value, } k)$$

The evaluation returns a Boolean value $\{True, False\}$. Intuitively, for a specified similarity function, the similarity predicate returns $True$ if the computed similarity score S between the input parameters satisfies $S \geq k$, or $False$ otherwise.

In our approach, *Similarity_Predicate_Function* specifies the similarity function used to compute the score *e.g., Jaro, Levenshtein, Jaccard, etc* . From an application point of view, it can be expressed in any object-relational database system as a UDF (*User Defined Function*)[Gra+01].

Now, consider each constraint $C \in Conds(q')$ of the query q' obtained in *step 1.* Let A be an attribute and v be a value such that both appear in C. If A appears in any attribute matching of Φ, then the constraint C is replaced by a similarity predicate where: The *Similarity_Predicate_Function* is the same similarity function in the attribute matching in Φ, the attribute A as the first parameter, the value v as the second and the threshold of the attribute matching as the last parameter.

Example 10. *Consider query q' (see example 9). Note that in $Conds(q')$ the attribute gender appears in the attribute matching $patient[sex] \approx_{(jaro,70)} donor[gender]$ of ϕ_{EM}. So, the constraint gender = 'M' is replaced by a jaro similarity function and its corresponding threshold.*

q' : **Select** donor_name,
donor_city,
blood_pressure,
blood_glucose, h_w
From donor
Where *jaro*(gender, 'M', 70)

7.4.1 | Correctness and completeness

In the following, we establish the correctness and the completeness of query translation. We provide the formal guarantees about our query translation algorithm.

At first, we give the correctness of query translation, then prove that our translation algorithms out put a correct translated query.

Definition 12. *Let Q and Q' be two queries over D and D', respectively, such that Φ is an entity matching rule between $R \in D$ and $R' \in D'$. We say that Q' is a sound correct translation of Q with respect to Φ , if for every $t \in Q(D)$ such that it there exists $t' \in D'$ where $t \sim_\Phi t'$, then $t' \in Q'(D')$.*

Theorem 1. *Our translation algorithm computes a correct translated query Q' from Q.*

Proof. Given a record $t \in D$ and a query Q over D such that:

$$t \in Q(D) \tag{7.1}$$

A record $t' \in D'$ such that:

$$t \sim_\Phi t' \tag{7.2}$$

From (7.1) we know that t satisfies $conds(Q)$.
From (7.2) we know that $t \sim_\Phi t' \iff t[A_1] \approx_1 t'[A\prime_1] \wedge \ldots \wedge t[A_n] \approx_n t'[A\prime_n]$ which means that t and t' satisfy the entity matching rule Φ.

$$\Phi \;=\; t\left[A_1\right] \approx_1 t'\left[A'_1\right] \wedge \ldots \wedge t\left[A_n\right] \approx_n t'\left[A'_n\right]$$

By definition, if Φ is *True*, then for $\forall p \in \Phi$ such that $p = \; t\left[A_i\right] \approx_i t'\left[A'_i\right], 1 \leq i \leq n$

$$p = True \tag{7.3}$$

If Q' is the translation query of Q, then $conds(Q')$ is a set of condition involving constraints obtained from $conds(Q)$.

Subsequently, $conds(Q')$ verifies the second translation condition (see Transaltion condition **??**), so $\forall\ C \in conds(Q')$, C is expressed as similarity predicate (see definition **??**) formed from a literal p, depending on the attribute in C (see step 2.).

Thus, from (7.1), (7.2), and (7.3), then t' satisfies each similarity predicate $C \in conds(Q')$. So, as t' satisfies each constraint $C \in conds(Q')$ then t' satisfies $conds(Q')$. Therefore, $t' \in Q'(D')$.

□

Next, we provide the formal definition of completeness and show that our translation algorithm output a unique translated query.

Definition 13. *Given Q and Q' be two queries over D and D', respectively, such that Φ is an entity matching rule between D and D'. We say that Q' is a complete translation of Q with respect to Φ , if Q' is a sound correct translation and for every sound correct translated query Q'' over D', $Q''(D) \subseteq Q'(D')$.*

Property 1. *From the definition 13, if two queries Q' and Q'' are complete translations of a query Q, then Q' and Q'' are equivalent.*

Theorem 2. *Our translation algorithm computes a complete translated query Q' from Q.*

Proof. Our translation is based on a unique entity matching rule that is a 1-1 mapping. Each constraint in Q is translated into a unique similarity predicate in Q'. □

At this stage of translation, q' is a well-formed query over the blood bank database. Next, we describe how the queries will be rewritten in such a way they preserves the access control.

7.5 | Query rewriting

Rewriting of the query is a key step in enforcing security policy of the data owners. It aims at returning only safe data in accordance with the defined access control. In this section, we will be focusing on how queries are evaluated with respect to a *security policy*. We remind that a set of access control rules denies access to an association of attributes under some constraints, as defined in the preliminary section.

Consider a query q over R with its set of access control rules $\Pi_R = \{r_1, ..., r_p\}$ with $p \geq 1$. We describe a query rewriting of q in such a way that the retrieved answers must not contain any tuple that could violate any rule of Π_R.

In the following we will introduce the definition of a critical tuple in order to explain the main idea behind our query rewriting policy preservation.

Definition 14. *(Critical Tuple) A tuple t from a given relation R is critical for a query q over R, if there exists a possible instance of the database I, which represents the actual content including the data itself at any given time of a database, where the presence or absence of t makes a difference to the result of q, i.e., $q(I \backslash \{t\}) \neq q(I)$.*

An access control rule is a view with the same syntax of an SPJ query (*see definition 9*).Thus, tuple t is sensitive when a given access control rule $r_i \in \Pi_R$ denies access to t. This means that t is a critical tuple for r_i.

Hence, a query q violates an access control rule r_i *iff* there exists a tuple $t \in R$ which is sensitive and critical for q. We can deduce that Π_R is violated *iff* the query q violates at least one rule of Π_R. In other words, a given query q violates an access control policy if there is a sensitive tuple critical for q.

Definition 15. *(Relevant Rule) Given a query q over R and a set of access control rules Π_R, a rule $r \in \Pi_R$ is relevant to a query q iff $Sel(r) \subseteq Sel(q)$.*

We note that not all rules in Π_R are relevant to q. Therefore, a set of relevant rules denoted Π_R^* must be constituted before the query is rewritten. Indeed, for each $r_i \in \Pi_R$ if $Sel(r_i) \subseteq Sel(q)$ then r_i is added to Π_R^*.

The basic idea behind building the set Π_R^* is to produce a query $q_{\Pi_R^*}$ which retrieves only authorization tuples. The query $q_{\Pi_R^*}$ considers the negation constraints of each rule in Π_R^* to get the following form of rewritten query:

$q_{\Pi^*_R}$: **Select** $Sel(\mathrm{q})$
From $tables(\mathrm{q})$
Where $Cond(\mathrm{q})$ **AND NOT** $Cond(r_1)$ **AND**
... **AND NOT** $Cond(r_m)$

Consequently, all the retrieved tuples are considered non-sensitive and do not violate any rule of Π^*_R are excluded from the result $q_{\Pi^*_R}$.

Example 11. *Consider a query q that retrieves name, city and blood glucose of all the male donors from the hospital data source:*

q : **Select** name_p , city_p , blood_glucose
From patient
Where sex='M'

For illustration purposes, we consider only this access control rule r_1 in the security policy $\Pi_{patient}$ of the hospital, which denies access to name and blood glucose for patients with diabetes as a disease.

r_1 : **Deny** name_p , blood_glucose
From patient
Where diagnosis='diabetes'

We construct the set $\Pi^*_{patient}$ which is composed of r_1 as $Sel(r_1) \subseteq Sel(q)$. Next, the rewritten query $q_{\Pi^*_{patient}}$ complying for $\Pi^*_{patient}$ that retrieves the name, city and the blood glucose of male patients without diabetes as a disease:

$q_{\Pi^*_1}$: **Select** name, city_p , blood_glucose
From patient
Where sex ='M' **AND NOT** disease='diabetes'

The evaluation of the rewritten query $q_{\Pi^*_{patient}}$ on the hospital instance of the motivating example (see the table 7.1) doesn't retrieve the patient 'Tim McCall' even his male. The rewritten exclude 'Tim McCall' because of the association name and blood glucose having a diagnosis diabetes disease, which is the main purpose of the access control rule r_1.

The rewriting query process allows a local database to grant strict access for retrieving safe records only. It considers the queries and access control rules inside the same source *i.e.,* from the same domain space. However, to return additional records from another database, a query posed over one database is augmented to another domain space through a translation process.

7.6 | Data sharing in compliance with access control policies

In this section, we describe the general framework where the different phases are put together. The main objectives of our work are to: (1) guarantee the preservation of the source policies and (2) guarantee that for each record t subject to an access control (AC), if t has an equivalent record t' in the other sources, then t' should comply with AC during the query evaluation process.

Definition 16. *(Data sharing in compliance with access control policies) Given a query q over a relation R, two access control policies Π_R and $\Pi_{R'}$ associated with R and R', respectively, and an entity matching rule Φ, a data sharing in accordance with access control policies should satisfy, for each tuple t in the answer to q:*

1. *Tuple t is not critical for any rule in Π_R.*
2. *If there exists $t' \in R'$ such as $t' \sim_\Phi t$ then t' is not a critical tuple for any rule in $\Pi_{R'}$.*

7.6.1 | General framework - naïve approach

Our approach requires a trusted third-party to handle the query answering process. Sources export their policy rules to the trusted third-party for policy rewriting. We assume that the sources do not try to compromise or collude with the third-party. The sources agree on the attribute alignment and the entity matching rules that will be used at the third-party. In the following, we describe, step by step, based on the previous definitions, how to answer a query q over a given source by extending vertically and horizontally its evaluation over other sources.

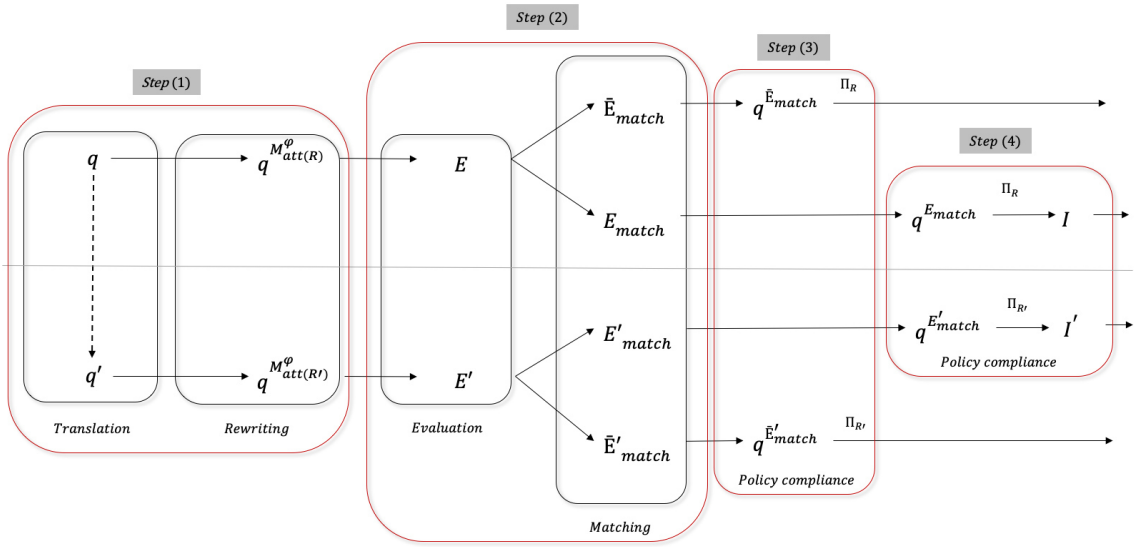


Figure 7.1: General framework for query evaluation in a data sharing setting

Step 1. This step is mainly based on returning the tuples from the sources aligned in such a way that we can perform entity matching comparisons. First, this consists in translating the query q over R into q' over R' as described in Section 7.4. Please note that the set of attributes $Sel(q)$ might not be suitable for entity matching since it needs all the attributes matching from both sources. Therefore, based on the query q, construct a query $q^{M^{\Phi}_{att(R)}}$ in the database D where the clauses have the following properties:

1. $Sel(q^{M^{\Phi}_{att(R)}}) = M^{\Phi}_{att(R)} \cup Sel(q)$. We remind you that $M^{\Phi}_{att(R)}$ is the set of attributes of R appearing in the matching rule $\mathbf{\Phi}$.
2. $Sel(q^{M^{\Phi}_{att(R)}}) = Conds(q)$.

The query $q^{M^{\Phi}_{att(R)}}$ take the following form:

$$q^{M^{\Phi}_{att(R)}}: \quad \begin{array}{ll} \textbf{Select} & M^{\Phi}_{att(R)} \cup Sel(q) \\ \textbf{From} & \text{Tables(q)} \\ \textbf{Where} & \text{Conds(q)} \end{array}$$

This operation is repeated to translated query q' in database D' to construct the query $q'^{M^{\Phi}_{att(R')}}$. The answer to $q^{M^{\Phi}_{att(R)}}$ and $q'^{M^{\Phi}_{att(R')}}$ will then be used to find matches according to $\mathbf{\Phi}$. Later on, the access control will be enforced through re writings in *step 3* (*see Figure 7.1*).

Step 2. In this step, we carry out, respectively, the evaluation of queries $q^{M^{\Phi}_{att(R)}}$ and $q'^{M^{\Phi}_{att(R')}}$ in the databases D and D'. Let E and E' be the results of $q^{M^{\Phi}_{att(R)}}$ and $q'^{M^{\Phi}_{att(R')}}$, respectively.

We apply the entity matching resolution to E and E' to identify the pairs of tuples that designate the same real-world object *w.r.t* the entity matching rule Φ_{EM}.

As a result, we obtain the set E_{match} (*resp.* E'_{match}) consisting of records in E that have a match in E' (*resp.* E). We also deduce $\overline{E}_{match}$ (*resp.* $\overline{E'}_{match}$) that contains the records in E (*resp.* E') for which there is no match.

Step 3. In this step, we deal with the sets, $\overline{E}_{match}$ and $\overline{E'}_{match}$. In each set, we remove the tuples that do not satisfy the local access control policies. First, we consider the set $\overline{E}_{match}$ of the non-matching records. We produce a query $q^{\overline{E}_{match}}$ of the following form:

$$q^{\overline{E}_{match}}: \quad \textbf{Select} \;\; Sel(q) \\ \textbf{From} \;\; \overline{E}_{match}$$

Then, $q^{\overline{E}_{match}}$ is evaluated in accordance with access control policy Π_R, following the query rewriting described in section 7.5. The result of this query evaluation will be part of the final answer. The same is done with set $\overline{E'}_{match}$.

Step 4. In this step, we consider the matching sets returned in *step 2*. For each set of tuples, we check the satisfiability of the access control policies.

Given the relation R and its corresponding access control policy Π_R, we apply Π_R to the returned matching records E_{match}. For that, we have to generate a query $q^{E_{match}}$ of the form:

$$q^{E_{match}}: \quad \textbf{Select} \quad * \\ \textbf{From} \quad E_{match}$$

After that, the query $q^{E_{match}}$ is rewritten in accordance with the access control Π_R. The query rewriting in this step has an exception compared to what we described in section 7.5. Thereby, in the query rewriting of $q^{E_{match}}$, the relevant rules are selected relatively to $Sel(q)$ instead of $Sel(q^{E_{match}})$.

The query evaluation of $q^{E_{match}}$ returns the set of tuples that we denote as $E^{\Pi_R}_{match}$, which contains only tuples that do not violate any access control rule Π_R.

The same operation is performed with the set E'_{match} as a result of which we obtain the set $E'^{\Pi_{R'}}_{match}$.

However, the set $E^{\Pi_R}_{match}$ may disclose some tuples that could violate the access control policy $\Pi_{R'}$. Indeed, some of the tuples in $E^{\Pi_R}_{match}$ may have a match in E'_{match} and do not appear in $E'^{\Pi_{R'}}_{match}$. In this case, they need to be removed from $E^{\Pi_R}_{match}$. Prior to this, we need to introduce the following definition.

Definition 17. *(Similarity-based Difference) [AH19] Given S and S', two relations with the same arity, and Φ an entity matching rule between S and S', the similarity-based difference, denoted $\underset{\Phi}{\setminus}$, is defined as: $S \underset{\Phi}{\setminus} S' = \{t |\ t \in S \wedge (\nexists\, t' \in S' : t \sim_\Phi t')\}$.*

Therefore, to return only safe tuples from source D, we use *Difference based on similarity* to calculate the final answer I as follows:

$$I = E^{\Pi_R}_{match} \underset{\Phi}{\setminus} (E'_{match} \setminus E'^{\Pi_{R'}}_{match})$$

The same operation is performed in D' to obtain I' (*see Figure 7.1*). Sets I and I' contain tuples that comply with access control of all sources. Relying on the attribute mapping of the setting, I and I' are projected on the attributes $Sel(q)$ and $Sel(q')$, respectively, then merged as a final result to query q.

Example 12. *Consider the query q of our motivating example. In step 1, q is translated into q' as seen in examples 9 and 10. Recall that the translated query q' obtained is as follow:*

q' :	**Select**	donor_name, donor_city,
		blood_pressure,
		blood_glucose, h_w
	From	donor
	Where	*jaro*(gender, 'M', 70)

Then always in step 1, construct a query $q^{M^\Phi_{att(patient)}}$ over the patient relation in the hospital database and $q'^{M^\Phi_{att(donors)}}$ over the donor relation in the blood bank database.

In step 2, the queries $q^{M^\Phi_{att(patient)}}$ and $q'^{M^\Phi_{att(donor)}}$ are evaluated in their respective database and lead to E and E' (see Tables 7.5 and 7.6):

At this stage we apply an entity matching process to identify the records that match according to Φ_{EM}. As a result, in one hand, we obtain E_{match} and E'_{match}

$q^{M^{\Phi}_{att(patient)}}$: **Select** name_p, address_p, city_p, sex, blood_pressure, blood_glucose, height_weight
From patient
Where sex = 'M'

$q'^{M^{\Phi}_{att(donors)}}$: **Select** donor_name, donor_address, donor_city, gender, blood_pressure, blood_glucose, h_w
From donor
Where *jaro*(gender, 'M', 70)

Table 7.5: The returned tuples E after the evaluation of $q^{M^{\Phi}_{att(patient)}}$

name_p	address_p	city_p	sex	blood_pressure	blood_glucose	height_weight
Bob Tracy	3 rue emile zola	lyonn	M	120/79	73	162/71
Smith, John	06 bis rue notre dame	paris 6	M	126/76	71	182/85
Tim McCall	43 av. des Postes	Lille center	M	124/75	131	175/42

that represent the sets of matching tuples. In the other hand, we obtain the sets $\overline{E}_{match}$ *and* $\overline{E'}_{match}$ *of the non-matching tuples. Please Note that in our scenario* $E_{match} = E$ *and* $E'_{match} = E'$*,*

Regarding the sets $\overline{E}_{match}$ *and* $\overline{E'}_{match}$ *both are empty in our case since all male patient are matching. Hence, the step 3 is skipped.*

Finally, in step 4 the goal is to remove from the answers of one source any tuple where its matching tuple in the other source is prohibited by policy. This step requires to consider the security policies attached to each database. We can notice that in the hospital database r_1 *and* r_2 *are not relevant to* q*. Therefore,* $E^{\Pi_{patient}}_{match} = E_{match}$*. In contrast, the rule* r'_1 *in the blood bank database is relevant to* $q'^{E_{match}}$ *because the attribute association between donor_name and blood_glucose of* $Sel(r'_1)$ *appears in* $Sel(q)$*. Hence, in the evaluation of* $q'^{E_{match}}$ *w.r.t the access control policy* Π_{donor} *the record corresponding "Timothy McCall", living in lille, is not returned, see Table 7.7.*

At this stage, now, we proceed to the removing of the tuples that have matching whose access is denied, from both of $E^{\Pi_{patient}}_{match}$ *and* $E'^{\Pi_{donor}}_{match}$*. We then compute two*

Table 7.6: The returned tuples E' after the evaluation of $q'^{M^{\Phi}_{att(donor)}}$

donor_name	donor_address	donor_city	gender	blood_pressure	blood_glucose	h_w
Robert Tracy	03 rue emile Zola	lyon	Male	120/79	71	162/72
John A. Smith	06 bis rue notre dame	paris	Male	127/77	70	181/89
Timothy McCall	43 aveunue des Postes	lille	Male	121/73	136	176/53

Table 7.7: Returned tuples $E'^{\Pi_{donor}}_{match}$

donor_name	donor_address	donor_city	gender	blood_pressure	blood_glucose	h_w
Robert Tracy	03 rue émile Zola	lyon	Male	120/79	71	162/72
John A. Smith	06 bis rue notre dame	paris	Male	127/77	70	181/89

sets, on one hand I as follow:

$$I = E^{\Pi_{patient}}_{match} \underset{\Phi}{\setminus} (E'_{match} \setminus E'^{\Pi_{donor}}_{match})$$

Note that the record corresponding to "Tim McCall" will be excluded from I since its match in $E'_{match} - E'^{\Pi_{donor}}_{match}$ is sensitive.

On the other hand:

$$I' = E'^{\Pi_{donor}}_{match} \underset{\Phi}{-} (E_{match} - E^{\Pi_{patient}}_{match})$$

where $I' = E'^{\Pi_{donor}}_{match}$ since $E_{match} - E^{\Pi_{patient}}_{match} = \emptyset$.

Table 7.8: Finale returned answer for Hospital database

name_p	city_p	blood_pressure	blood_glucose	height_weight
Bob Tracy	lyonn	120/79	73	162/71
Smith, John	paris 6	126/76	71	182/85

Table 7.9: Finale returned answer for Blood Bank database

donor_name	donor_city	blood_pressure	blood_glucose	h_w
Robert Tracy	lyon	120/79	71	162/72
John A. Smith	paris	127/77	70	181/89

The final answers are shown in Tables 7.8 and 7.9 and they are a projection of I and I' on $sel(q)$ and $sel(q')$, respectively.

The approach presented above requires all participating sources to trust the third-party. Trust involves the sources exposing their access control to the third-party. However, it is a naive approach that exempts the sources from query evaluation since the entire process is handled by the third-party.

7.6.2 | Hiding Access control policy

Access control policy rules are themselves knowledge for an attacker that could be used to infer sensitive information [FAL06]. Considering access policies to be sensitive information and hide them, is not only for a malicious user, but, in some cases, it could be a commercial secret (*i.e.,* the access control policy could disclose business strategies) that could compromise the strategy of the owner.

In the following, we describe a configuration - *figure 7.2* - that would achieve data sharing in compliance with access control policies without revealing the access control to the third party[**agoun2021data**].

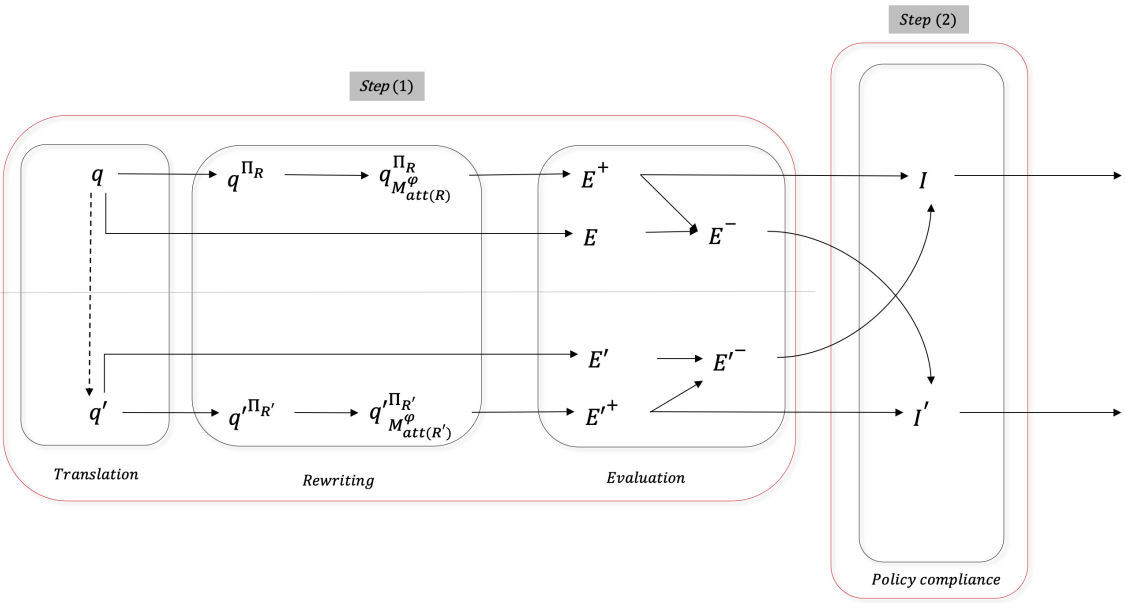


Figure 7.2: General framework for query evaluation in a data sharing setting without revealing the access control policies

Step 1. Similarly to *step 1* in section 7.6.1, the database D, receiving the query q over R, translates q into q' over R' in D'. Then, instead of constructing the query $q^{M^{\Phi}_{att(R)}}$ it proceeds to rewrite q in accordance with access control Π_R as described in *section 6*. The resulting query is denoted q_{Π_R}. Before evaluating q_{Π_R}, the set of matching attributes $M^{\Phi}_{att(R)}$ is unified with $Sel(q_{\Pi_R})$ and produces the query that we denote $q^{M^{\Phi}_{att(R)}}_{\Pi_R}$. The set of returned tuples is denoted E^+ and represents the set of authorized records. The originality of this approach is that it identifies the set of sensitive tuples E^- of q in local. However, we need to retrieve all tuples of q without enforcing access control: we denote this set by E. Then, E^- is derived from E as follows:

$$E^- = E \backslash E^+$$

The database D' proceeds in like manner with the translated query q' to compute E'^+ and E'^-, respectively. Then each of the sources send the resulting sets to the trusted third-party.

Step 2. In this step, the third-party handles the process to ensure access control policy compliance between the sources. Thus, relying on the similarity-based difference, (see in *Definition 17*), the sensitive tuples of the database D' that have a match inside the set E^+ are removed. This is computed as:

$$I = E^+ \underset{\Phi}{\backslash} E'^-$$

The same operation is performed in database D to compute I'. Finally, we prepare the final answer by returning the merged projection of I and I' on the attribute $Sel(q)$ and $Sel(q')$, respectively.

Since we assumed that the third-party is trusted by the participating sources, it was possible to perform access control compliance.

7.7 | Implementation

We implemented our approach according to an architecture based on a trusted third-party. We assume that a trusted third-party does not collude with any site to violate the security policies.

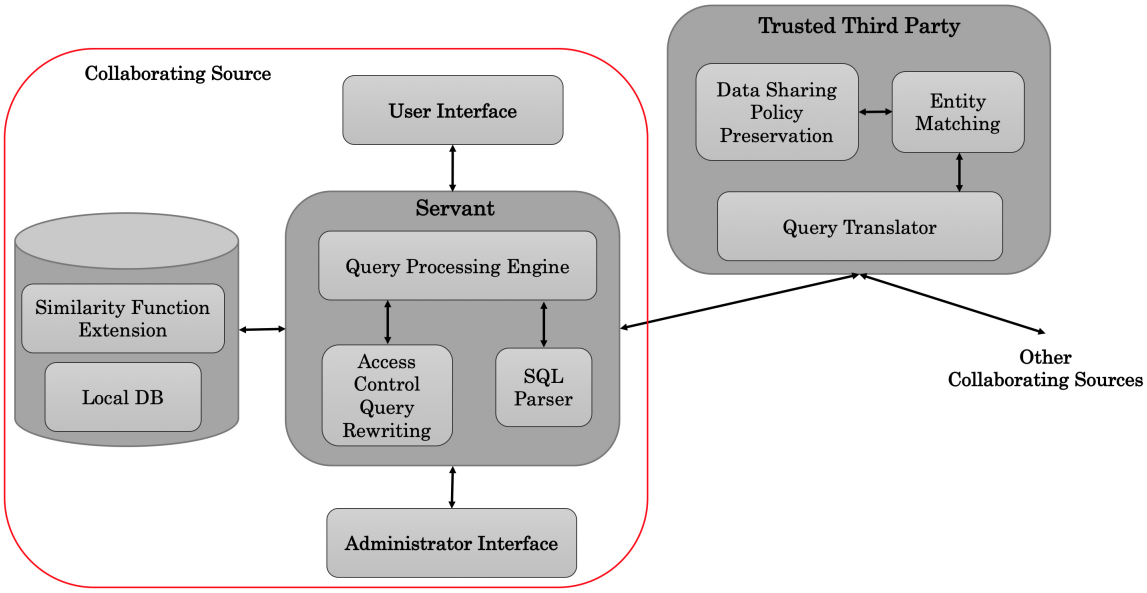


Figure 7.3: Data sharing architecture with access control policy preservation

The structure of each involved site is shown in *Figure 7.3*. We provide interfaces through which a user could issue queries, and an administrator set up the access control rules. Each collaborating source manages its own collection of data and autonomously chooses its logical database design. We chose *PostgreSQL 11.4* relational DBMS as a data store since it allows extensions to be integrated into the database (*e.g.,* external functions). To support calls to similarity functions in queries, we added to each RDBMS source the *pg_similarity*[2] extension that defines similarity functions in addition to the traditional operators. These functions can be used as UDFs to implement similarity algorithms available in the literature.

The main component in each collaborating site is the servant. It consists of three modules: (*i*) The Query Processing Engine is the core of the system that manages the query exchanges in each source. (*ii*) The SQL Parser that checks the query form, the ability for evaluation, and whether it is possible to translate it. (*iii*) The access control query rewriting module implementing the algorithms that enforce the local access control policy.

The third-party is in charge of supervising the data sharing process in accordance with the access control policies of the collaborating sources. It is composed of three main modules: (*i*) The Entity Matching module is a library *PyRLT*[3] intended to

[2]https://github.com/eulerto/pg_similarity
[3]https://recordlinkage.readthedocs.io/en/latest/about.html

link records within or between two data sets. It implements indexing methods and functions to compare records based on the entity matching rules. It also collaborates with the Data Sharing Policy Preservation module to compute the similarity-based difference. (*ii*) The Query Translator module implements the algorithm devoted to query translation: Given a query q and an entity matching rule Φ with attribute alignment m, it translates q over a source into an equivalent query q' over another source. (*iii*) The Data Sharing Policy Preservation module in charge of removing records that violate the access control policy of the other sources.

The servant and the trusted third-party were implemented in *Python 3.6.9* and it contains approximately two thousands lines of source code. Figure 7.3 describes the architecture adapted to data sharing based on entity matching rules with hidden access control rules.

7.8 | Experiments

For the evaluation of our algorithms, we undertook two studies. The objective of the first study was to investigate the time performance of our algorithms with respect to two parameters, namely, size of the databases and the number of sensitive records. The second study investigates the effectiveness of our data sharing approach. We conducted experiments on both real and synthetic data sets. *Restaurant*[4] is the real data set we used and is a collection of 858 distinct records of 753 restaurants. Each record has four attributes: *name, address, city,* and *type*. This data set identifies 106 positive pairs over 184,041 pairs. The sources were generated synthetically using the real data set for the first study, whereas in the second study, the experiment was conducted only on the real data.

We used the real data to generate two distinct sources and populated each source for one of our experimental studies.

To avoid network latency and measure only the performance of our algorithms, all the experiments are run in a single machine equipped with 2.2 GHz Quad-Core Intel Core *i7* processor and 16 GB 1600 MHz DDR3 RAM. The experimental setup consists of two postgreSQL 11.4 servers hosting the data sources and the trusted third-party module with the structure shown in *figure 7.3*.

[4]https://www.cs.utexas.edu/users/ml/riddle/data/restaurant.tar.gz

7.8.1 | Execution Time

Given a query Q submitted to the data sharing platform with policy preservation, we expect that the time to answer Q is influenced by the presence of large data sets of records in the sources, due to the entity matching configuration. Therefore, the fundamental questions we address are: (i) How does our approach behave when the number of records increases? (i) What is the exact relationship between time and size?

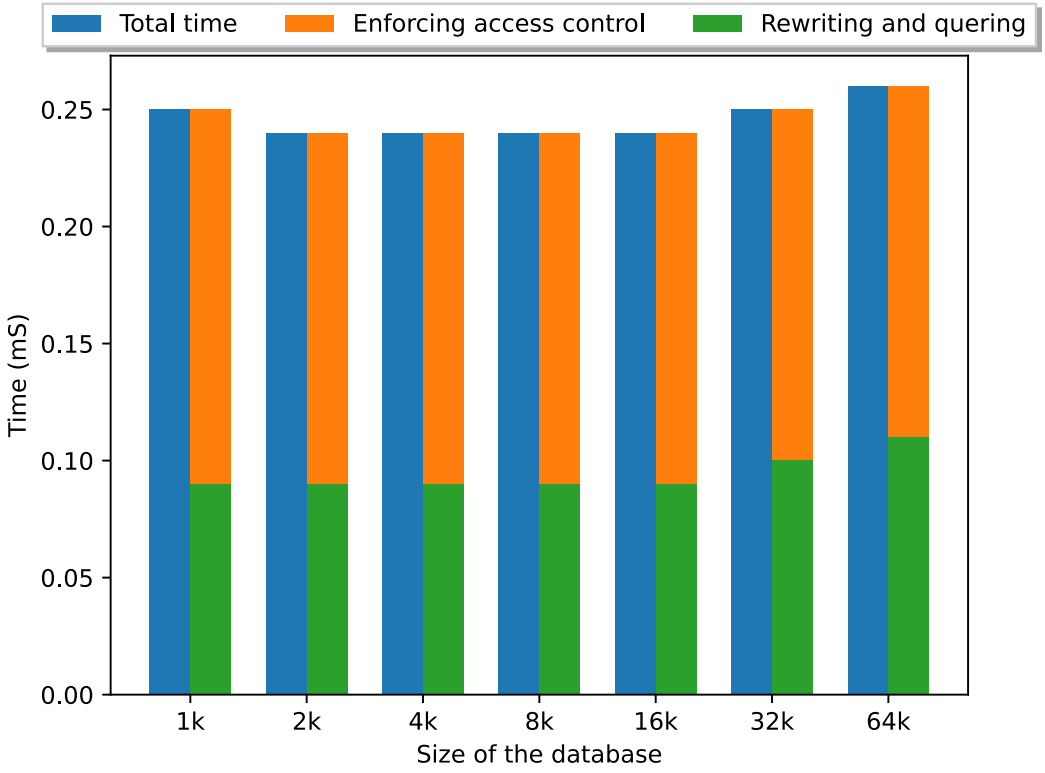


Figure 7.4: Average query execution time comparison for different database sizes

Regarding the first question, we selected 5 distinct queries, each of which is executed several times by enforcing the same security policy. The queries are selected in such a manner that they have approximately the same number of sensitive records. The security policies are not the same for both sources but they do not change when the number of records is increased. The objective is to measure query evaluation time when record size is increase. Figure 7.4 shows the time breakdown of query evaluation. The first column shows the average time of a query evaluation, the

second one shows the average time for rewriting and querying, and the third one shows the average time for enforcing compliance with access control policies between the two sources. As translation time was very negligible, we chose not to represent it. The results show that the average running time is the same for those queries. As we can see, increasing the number of records in both sources does not seem to greatly influence time even if we can notice a slight increase in rewriting and querying time.

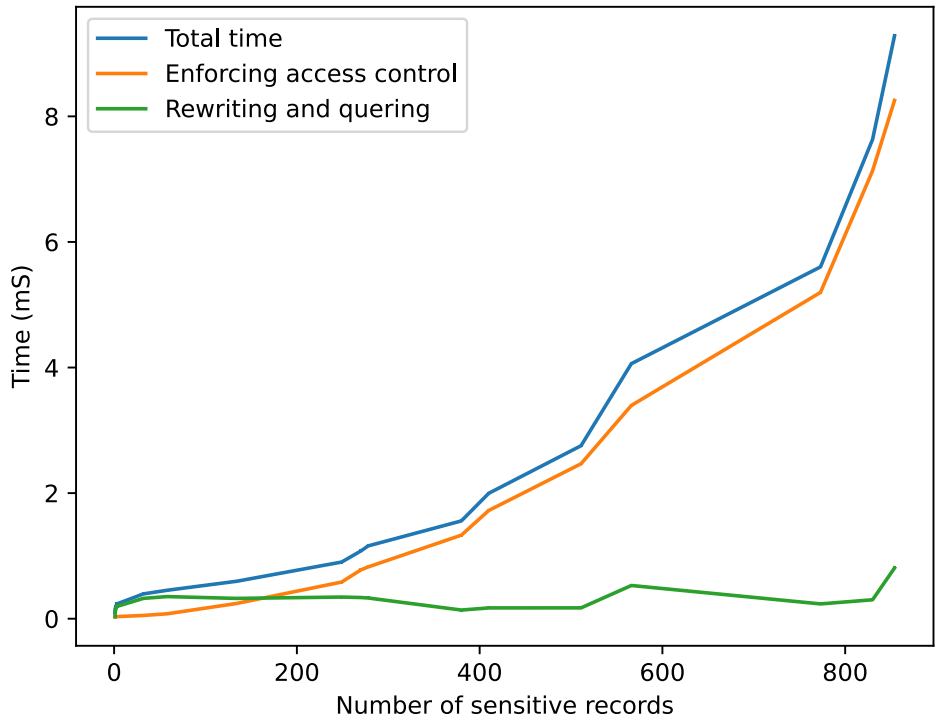


Figure 7.5: Average query execution time comparison as the number of sensitive records increases

In the previous experiment, the input queries have the same number of sensitive records. In this experiment, we adopt a fixed number of records in both sources to $(2k)$ and increase the number of retrieved records in such a way as to vary the number of sensitive records in sources. We select 20 distinct queries each of which, the WHERE clause increases the number of disjuncts. We select 20 distinct queries in such a way as the number of disjunctions in the where clause is not the same. For example, the first query has 2 disjunctions of predicates, the second has 4

disjunctions, etc. It leads to an increase in the number of retrieved records from both sources. The result is an increase in the number of sensitive records.

Figure 7.5 shows curves for the total running time of query evaluation, the running time for rewriting and querying, and running for the time required by policy enforcement with access control policies. Analyzing the curves, we notice that from 200 sensitive records, the running time scales gracefully which gives to the curve a quadratic shape. Entity matching complexity is $O(n^2)$ since it requires N^2 comparison, where N is the size of the data sets. Note that we considered the worst case performance. The increase in sensitive records means that query evaluation is time-consuming. Thus, total time is greatly influenced by the time required to enforce access control policies.

7.8.2 | Effectiveness

The key question we answer in this section is: How effective is our data sharing approach in presence of the access control policies?

To investigate the effectiveness of our algorithms, we conducted an experiment on the real data set *Restaurant*. We generated two databases, each with 429 records where 106 are matching bases on the data set *Restaurant*. Then, we selected 30 distinct queries in each of which the size of the disjuncts in the WHERE clause varies from 1 to 30. Thanks to Wang et al. [Wan+11a] we obtained their implementation to find the appropriate entity matching rule. This technique requires a set of examples as inputs, it includes positive examples, known to be the same entity, and negative examples, known not to be the same entity. In this experiment, we fix the size of negative examples and vary the number of positive examples. Then we construct a matching rule on three attributes: *name*, *address* and *city*. The positive examples are selection randomly from the 106 matching pairs. We run the discovering rule 10 times and, for each entity matching rule performed, we run the queries and measure the average precision and recall. Based on the ground truth data in *Restaurant*, our precision calculates the proportion of how many of the returned records have were safe. On the other hand, the recall measures how correctly are the retrieved record are, referring the ground truth data. Figure 7.6 shows the average of precision, recall, and the F-measure values of the input queries by varying the size of positive examples. We can see that the precision scales smoothly from 94% to 96% while the recall is always at 1.0. We explain these results by the

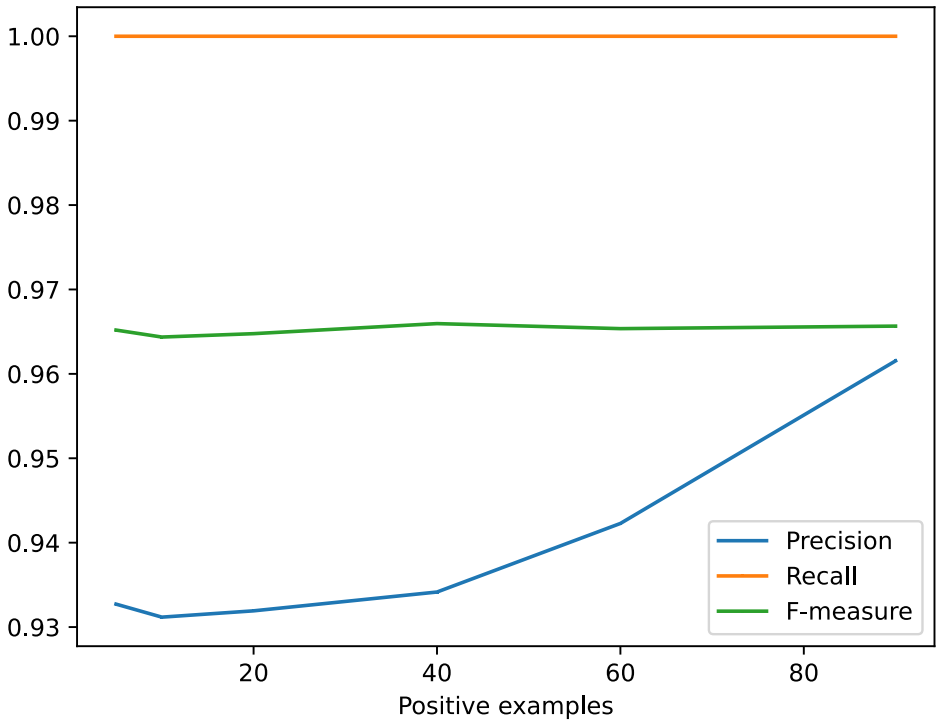


Figure 7.6: Effectiveness results using different positive examples

fact that our approach retrieves the true matches and applies a strict elimination of sensitive records. Nonetheless, some of the pairs are not classified as matches since they do not satisfy the entity matching rules. On average, our approach achieves an F-measure of 97%. Adopting an appropriate similarity function and thresholds guarantees high precision, particularly, to find the records that match and are leaked by one of the sources.

We also experimented our approach by duplicating one of the sources. Therefore, using the exact similarity function inside the matching rule, we found an outperformed result with an F-measure of 1.0.

7.9 | Summary

In this chapter, we considered the problem of data sharing between heterogeneous and autonomous data sources in the presence of access control policies. We de-

scribed an approach where several sources could share their data according to their security policies. We used entity matching rules to link data from different sources and we showed how the entity matching rules are used in the translation of structured queries. We introduced the notion of data sharing compliance with access control and how to handle the preservation of all security policies. We aimed at a restrictive approach, in the sense that, in event of conflict, denial access to data takes precedence. Reference architecture requires a trusted third-party to manage the query translation and prevent disclosure of sensitive data.

The experimental study shows that our solution follows a quadratic scale depending on the number of sensitive records in the databases, because of the pairwise comparison to compute the similarity. While enforcement of security policies is mainly based on entity matching processes, it has become problematic for very large data sources. We are aware that when the size of the sensitive records is in millions or billions, it will become problematic. However, in our solution the pairwise matching concerns only sensitive records of both sources, which are much less comparing to the whole database. We aim to preserve the security policies in such a data sharing setting, not to propose a more efficient record matching. However, techniques such as blocking aim at reducing comparison search space thus avoiding comparing all pair of records.

8

Conclusions and Perspectives

Contents

With the ubiquity of social networks in modern life, users often post their activities and by what reveal details of their privacy. The interest in identifying multiple pieces of information of the same individual, important for analysis, becomes a booming business. For instance, companies are interested in correlating a user's activities to gather all the information across multiple platforms (such as data brokers, social media, etc.) and build a more complete profile of an individual. Asserting rights of information usage after providing a digital platform becomes a confusing task when their number increases. With this research, we attempt to draw attention about the risk of mass data sharing where even more overlapping of data is expected.

8.1 | Summary

In this thesis, we investigated the problem of data sharing between autonomous sources while preserving security policies. Enforcing access control policies in a data sharing system is a challenging task that became complex when there is overlapping information. We focused on the flaw that might occur when a same real-world entity is derived from different data sources with conflicting access restrictions. We started by a wide-ranging review of different fields relevant to our studied issue, namely, data interoperability, access control, data matching. Based on the lessons learned from the literature review, we have formalized this problem.

Our first objective targeted on the role of the interaction between all elements of a data sharing system mainly the local policies, the local sources and entity matching rules. We considered an instance-oriented approach to enable sources to share their data through entity matching rules to specify mappings. In this context, our aim is to allow a query user to retrieve an enriched answer from external sources without violating any security policy. To enforce the security and maximize sharing we considered two distinct levels, namely, local and global.

First, we have studied the problem of data publishing in presence of security policies. Before sharing, the data owner sets up a policy requirement to describe the part of the information to keep secret. Our goal from preserving the sources was not always to apply a strong restriction, but to find a balance between data security and data availability. To achieve this goal, we have proposed an approach for revising publication views highlighted as follows:

- We described, through a concept of disjoint queries, how to detect, from the set of views, those that violate the access control rules. We have considered that when a publication view and the policy rule have results in common there is a violation.
- For the views that do not preserve the security policies, we provided an approach based on the concept of residue usually adopted in semantic query optimization. The views are rewritten using the constraints of the violated policy rules. This process helps to provide more fine-grained restrictions by targeting only tuples that are raise security issues.

Secondly, we have considered the problem of sharing data between two sources where the policy, database instance and schema might change. We have demon-

strated through an illustrative example that traditional access controls are not always efficient and could not prevent the disclosure of sensitive information in such a context. We have provided a practical methodology for data sharing based on data-instances that preserves the local access control policies. We made the following contributions:

- We introduced a formal definition about query answering in data sharing. We present a technique for translating queries using entity matching rules. We introduced the definition of correctness and completeness of query translation then proved that our translation algorithm computes a correct and complete query.
- We provided a definition of data sharing compliance with access control and we proposed how to handle the preservation of all security policies. We introduced an online query rewriting to enforce local access control policies inside each of the collaborating sources.
- We have presented two detailed strategies to achieve query answering in data sharing complying with local access control policies, involving a trusted third party. In the first strategy, the third party manages the entire process, offering a flexible and fine-grained access control. In the second strategy, the access control is enforced without the policies being exposed to the third party, preventing the restriction mechanism from being figured out.
- The implementation of our framework showed that our methodology achieves the security objectives when the rules are increasingly appropriate to the handled data. The execution time is mostly influenced by execution time of the entity matching process. Although the execution time takes a quadratic evolution without any optimisation process (e.g., blocking), it is only a worst-case complexity, which does not happen often.

8.2 | Future work

There are many research directions to pursue:

- Accommodating our approach with data dependencies and complex interactions between views would be an interesting extension. Indeed, when there

are multiple published views, a violation could potentially occur from a combination of views which, individually, are privacy-preserving *w.r.t.* a set of policy rules. We could pursue our revising process by considering the interplay between multiple views.

- We have considered access control rules as forbidden views that deny access to a portion of data. However, we could consider access control that both allows and denies access to specific information using environment parameters (time, location, etc.). Investigating such hybrid access control principles will allow data owner be more flexible to set up their security policy.
- The framework can be extended with the possibility of handling more complex entity matching rules. Actually, we could involve an enriched grammar, General Boolean Formula (GBF) to capture high-level specifications. Expression entity matching rules with GBF will allow handling missing values. EM rules in the form of GBF are more concise since it combines conjunctions ($\wedge$), disjunctions ($\vee$) and negations ($\neg$).
- Retrieving results from multiple data sources involves the use of data fusion technique to obtain a unified representation. An important area to explore is the data fusion. The problem to address would be how to consolidate all the returned records? Particularly those of the same real-world entities having multiple representations. [Ben+19].
- A decentralized runtime system would reduce the reticence of data owners for sharing their data. In fact, a mechanism without a shared component would strengthen our proposed approach (see figure 8.1). Sources will avoid outsourcing their data and will have better control over them. Eventually, it would be interesting to consider an architecture without any third-party. This perspective mostly relies on Secure Multi-party Computation which achieves query computation by keeping data private (i.e., encrypted) [Cos+18]. Figure 8.1, illustrates an architecture that allows to augment query result q of source 1 with those in source 2 without relying on a third party. Though, the drawback of such a solution is needs to exchange of results (i.e., records that preserve local access control policies) with the other source,which may become impractical as the size of the query results increases.
- Involve more than two data sources. In this vision, several issues need to be designed and evaluated, for instance, the optimization of some crucial steps

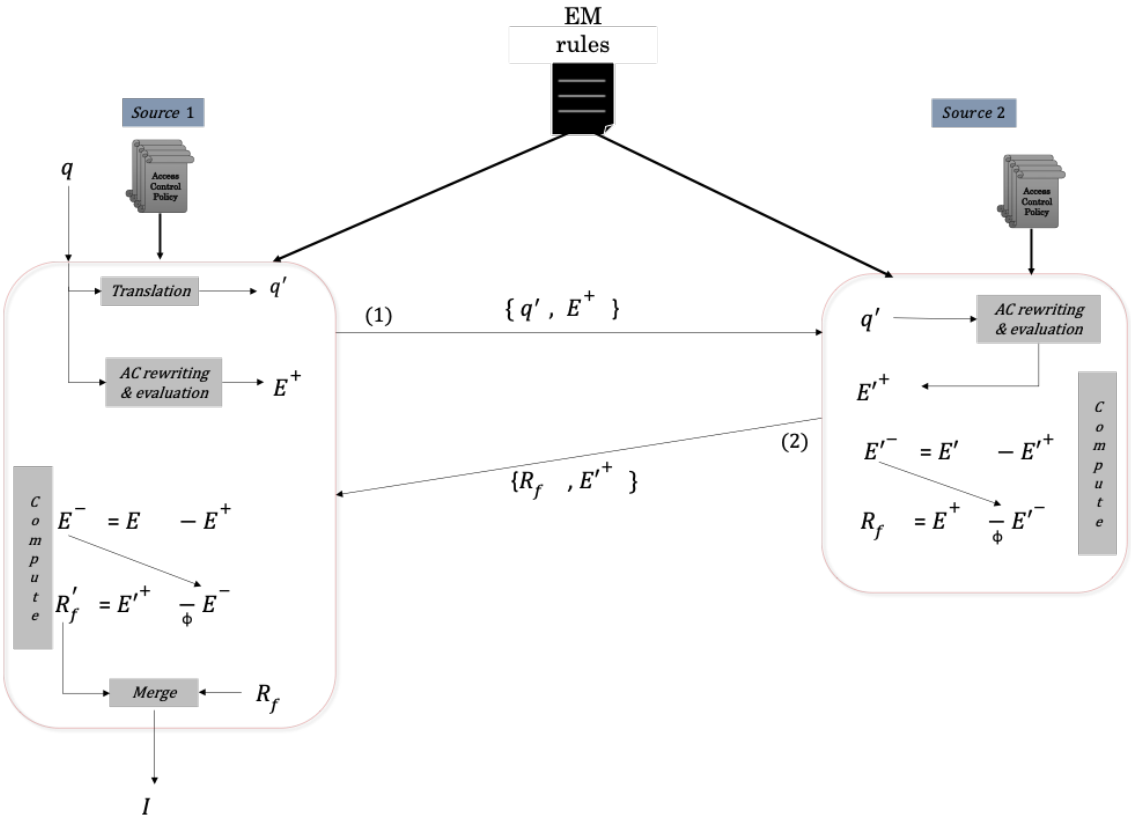


Figure 8.1: Secure decentralised data sharing system

like finding matching pairs and removing sensitive matches. This could also be optimized by extending our solution with learning approaches to find e.g., more appropriate entity matching rules.

References

[Dun46] Halbert L Dunn. “Record linkage”. In: *American Journal of Public Health and the Nations Health* 36.12 (1946), pp. 1412–1416.

[FS69] Ivan P Fellegi and Alan B Sunter. “A theory for record linkage”. In: *Journal of the American Statistical Association* 64.328 (1969), pp. 1183–1210.

[Lam74] Butler W Lampson. “Protection”. In: *ACM SIGOPS Operating Systems Review* 8.1 (1974), pp. 18–24.

[Che76] Peter Pin-Shan Chen. “The entity-relationship model—toward a unified view of data”. In: *ACM transactions on database systems (TODS)* 1.1 (1976), pp. 9–36.

[GW76] Patricia P Griffiths and Bradford W Wade. “An authorization mechanism for a relational database system”. In: *ACM Transactions on Database Systems (TODS)* 1.3 (1976), pp. 242–255.

[HRU76] Michael A Harrison, Walter L Ruzzo, and Jeffrey D Ullman. “Protection in operating systems”. In: *Communications of the ACM* 19.8 (1976), pp. 461–471.

[Fag78] Ronald Fagin. “On an authorization mechanism”. In: *ACM Transactions on Database Systems (TODS)* 3.3 (1978), pp. 310–319.

[HM78] Michael Hammer and Dennis McLeod. “The Semantic Data Model: A Modelling Mechanism for Data Base Applications”. In: *Proceedings of the 1978 ACM SIGMOD International Conference on Management of Data*. SIGMOD ’78. Austin, Texas: Association for Computing Ma-

chinery, 1978, pp. 26–36. ISBN: 9781450373425. DOI: 10.1145/509252.509264. URL: https://doi.org/10.1145/509252.509264.

[DJL79] David Dobkin, Anita K Jones, and Richard J Lipton. "Secure databases: Protection against user influence". In: *ACM Transactions on Database systems (TODS)* 4.1 (1979), pp. 97–106.

[Den80] Dorothy E. Denning. "Secure Statistical Databases with Random Sample Queries". In: *ACM Trans. Database Syst.* 5.3 (Sept. 1980), pp. 291–315. ISSN: 0362-5915. DOI: 10.1145/320613.320616. URL: https://doi.org/10.1145/320613.320616.

[HD80] Patrick AV Hall and Geoff R Dowling. "Approximate string matching". In: *ACM computing surveys (CSUR)* 12.4 (1980), pp. 381–402.

[MB81] Amihai Motro and Peter Buneman. "Constructing superviews". In: *Proceedings of the 1981 ACM SIGMOD international conference on Management of data*. 1981, pp. 56–64.

[LCL85] Chong K Liew, Uinam J Choi, and Chung J Liew. "A data distortion by probability distribution". In: *ACM Transactions on Database Systems (TODS)* 10.3 (1985), pp. 395–411.

[SO87] Tzong-An Su and Gultekin Ozsoyoglu. "Data dependencies and inference control in multilevel relational database systems". In: *1987 IEEE Symposium on Security and Privacy*. IEEE. 1987, pp. 202–202.

[MJ88] Catherine Meadows and Sushil Jajodia. "Integrity versus security in multi-level secure databases". In: *on Database Security: Status and Prospects*. 1988, pp. 89–101.

[Mor88] Matthew Morgenstern. "Controlling logical inference in multilevel database systems". In: *Proceedings. 1988 IEEE Symposium on Security and Privacy*. IEEE Computer Society. 1988, pp. 245–245.

[AW89] Nabil R Adam and John C Worthmann. "Security-control methods for statistical databases: a comparative study". In: *ACM Computing Surveys (CSUR)* 21.4 (1989), pp. 515–556.

[Mot89] A. Motro. “An access authorization model for relational databases based on algebraic manipulation of view definitions”. In: *[1989] Proceedings. Fifth International Conference on Data Engineering*. 1989, pp. 339–347. DOI: 10.1109/ICDE.1989.47234.

[CGM90] Upen S Chakravarthy, John Grant, and Jack Minker. “Logic-based approach to semantic query optimization”. In: *ACM Transactions on Database Systems (TODS)* 15.2 (1990), pp. 162–207.

[OS90] Gultekin Ozsoyoglu and Tzong-An Su. “On inference control in semantic data models for statistical databases”. In: *Journal of Computer and System Sciences* 40.3 (1990), pp. 405–443.

[SL90] Amit P Sheth and James A Larson. “Federated database systems for managing distributed, heterogeneous, and autonomous databases”. In: *ACM Computing Surveys (CSUR)* 22.3 (1990), pp. 183–236.

[WS92] W John Wilbur and Karl Sirotkin. “The automatic identification of stop words”. In: *Journal of information science* 18.1 (1992), pp. 45–55.

[Cha+94] Sudarshan Chawathe et al. “The TSIMMIS project: Integration of heterogenous information sources”. In: (1994).

[BJS95] Elisa Bertino, Sushil Jajodia, and Pierangela Samarati. “Database security: research and practice”. In: *Information systems* 20.7 (1995), pp. 537–556.

[FCK95] David Ferraiolo, Janet Cugini, and D Richard Kuhn. “Role-based access control (RBAC): Features and motivations”. In: *Proceedings of 11th annual computer security application conference*. 1995, pp. 241–48.

[JM95] Sushil Jajodia and Catherine Meadows. “Inference problems in multilevel secure database management systems”. In: *Information Security: An integrated collection of essays* 1 (1995), pp. 570–584.

[DH96] Harry S. Delugach and Thomas H. Hinke. “Wizard: A database inference analysis and detection system”. In: *IEEE Transactions on Knowledge and Data Engineering* 8.1 (1996), pp. 56–66.

[Qia96] Xiaolei Qian. “View-based access control with high assurance”. In: *Proceedings 1996 IEEE Symposium on Security and Privacy*. IEEE. 1996, pp. 85–93.

[SS96] Ravi Sandhu and Pierangela Samarati. “Authentication, access control, and audit”. In: *ACM Computing Surveys (CSUR)* 28.1 (1996), pp. 241–243.

[DS97] Sabrina De Capitani di Vimercati and Pierangela Samarati. “Authorization specification and enforcement in federated database systems”. In: *Journal of Computer Security* 5.2 (1997), pp. 155–188.

[HDW97] Thomas H Hinke, Harry S Delugach, and Randall P Wolf. “Protecting databases from inference attacks”. In: *Computers & Security* 16.8 (1997), pp. 687–708.

[Coh98] William W Cohen. “Integration of heterogeneous databases without common domains using queries based on textual similarity”. In: *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*. 1998, pp. 201–212.

[Par98] Donn B Parker. *Fighting computer crime: A new framework for protecting information*. John Wiley & Sons, Inc., 1998.

[San98] Ravi S Sandhu. “Role-based access control”. In: *Advances in computers*. Vol. 46. Elsevier, 1998, pp. 237–286.

[CG99] Frédéric Cuppens and Alban Gabillon. “Logical foundations of multilevel databases”. In: *Data & Knowledge Engineering* 29.3 (1999), pp. 259–291.

[FBK99] David F Ferraiolo, John F Barkley, and D Richard Kuhn. “A role-based access control model and reference implementation within a corporate intranet”. In: *ACM Transactions on Information and System Security (TISSEC)* 2.1 (1999), pp. 34–64.

[FLM+99] Marc Friedman, Alon Y Levy, Todd D Millstein, et al. “Navigational plans for data integration”. In: *AAAI/IAAI* 1999 (1999), pp. 67–73.

[SL99] Shiuh-Pyng Shieh and Chern-Tang Lin. “Auditing user queries in dynamic statistical databases”. In: *Information sciences* 113.1-2 (1999), pp. 131–146.

[BS00] Piero Bonatti and Pierangela Samarati. “Regulating service access and information release on the web”. In: *Proceedings of the 7th ACM conference on Computer and communications security*. 2000, pp. 134–143.

[BFJ00] Alexander Brodsky, Csilla Farkas, and Sushil Jajodia. “Secure databases: Constraints, inference channels, and monitoring disclosures”. In: *IEEE Transactions on Knowledge and Data Engineering* 12.6 (2000), pp. 900–919.

[Coh00] William W Cohen. “Data integration using similarity joins and a word-based information representation language”. In: *ACM Transactions on Information Systems (TOIS)* 18.3 (2000), pp. 288–321.

[Gra+00] John Grant et al. “Logic-based query optimization for object databases”. In: *Knowledge and Data Engineering, IEEE Transactions on* 12 (Aug. 2000), pp. 529–547. DOI: 10.1109/69.868906.

[MNU00] Andrew McCallum, Kamal Nigam, and Lyle H Ungar. “Efficient clustering of high-dimensional data sets with application to reference matching”. In: *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2000, pp. 169–178.

[MHH00] Renée J Miller, Laura M Haas, and Mauricio A Hernández. “Schema mapping as query discovery”. In: *VLDB*. Vol. 2000. 2000, pp. 77–88.

[Mon00] Alvaro E. Monge. “Matching algorithms within a duplicate detection system”. In: *IEEE Data Eng. Bull.* 23.4 (2000), pp. 14–20.

[PL00] Rachel Pottinger and Alon Levy. “A scalable algorithm for answering queries using views”. In: *VLDB*. 2000, pp. 484–495.

[RS00] Arnon Rosenthal and Edward Sciore. “View security as the basis for data warehouse security.” In: Jan. 2000, p. 8.

[SV00] Pierangela Samarati and Sabrina Capitani de Vimercati. “Access control: Policies, models, and mechanisms”. In: *International School on Foundations of Security Analysis and Design*. Springer. 2000, pp. 137–196.

[Ull00] Jeffrey D Ullman. “Information integration using logical views”. In: *Theoretical Computer Science* 239.2 (2000), pp. 189–210.

[Fer+01] David F Ferraiolo et al. “Proposed NIST standard for role-based access control”. In: *ACM Transactions on Information and System Security (TISSEC)* 4.3 (2001), pp. 224–274.

[For+01] Marco Fortini et al. “On Bayesian record linkage”. In: *Research in Official Statistics* 4.1 (2001), pp. 185–198.

[Gil01] Leicester Gill. *Methods for automatic record matching and linkage and their use in national statistics*. 25. Office for National Statistics, 2001.

[Gra+01] Luis Gravano et al. “Approximate string joins in a database (almost) for free”. In: *VLDB*. Vol. 1. 2001, pp. 491–500.

[Hal01] Alon Y Halevy. “Answering queries using views: A survey”. In: *The VLDB Journal* 10.4 (2001), pp. 270–294.

[LWO01] Weifa Liang, Hui Wang, and Maria E Orlowska. “Materialized view selection under the maintenance time constraint”. In: *Data & Knowledge Engineering* 37.2 (2001), pp. 203–216.

[RB01] Erhard Rahm and Philip A Bernstein. “A survey of approaches to automatic schema matching”. In: *the VLDB Journal* 10.4 (2001), pp. 334–350.

[Chu+02] Tim Churches et al. “Preparation of name and address data for record linkage using hidden Markov models”. In: *BMC Medical Informatics and Decision Making* 2.1 (2002), pp. 1–16.

[EVE02] Mohamed G Elfeky, Vassilios S Verykios, and Ahmed K Elmagarmid. “TAILOR: A record linkage toolbox”. In: *Proceedings 18th International Conference on Data Engineering*. IEEE. 2002, pp. 17–28.

[FJ02] Csilla Farkas and Sushil Jajodia. “The inference problem: A survey”. In: *ACM SIGKDD Explorations Newsletter* 4.2 (2002), pp. 6–11.

[Len02] Maurizio Lenzerini. “Data integration: A theoretical perspective”. In: *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM. 2002, pp. 233–246.

[LWJ02] Yingjiu Li, Lingyu Wang, and Sushil Jajodia. “Preventing interval-based inference by random data perturbation”. In: *International Workshop on Privacy Enhancing Technologies*. Springer. 2002, pp. 160–170.

[MMH02] Lik Mui, Mojdeh Mohtashemi, and Ari Halberstadt. “A computational model of trust and reputation”. In: *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*. IEEE. 2002, pp. 2431–2439.

[OCM02] Michael Ortega-Binderberger, Kaushik Chakrabarti, and Sharad Mehrotra. “An approach to integrating query refinement in SQL”. In: *International Conference on Extending Database Technology*. Springer. 2002, pp. 15–33.

[Swe02] Latanya Sweeney. “k-anonymity: A model for protecting privacy”. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.05 (2002), pp. 557–570.

[TKM02] Sheila Tejada, Craig A Knoblock, and Steven Minton. “Learning domain-independent string transformation weights for high accuracy object identification”. In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2002, pp. 350–359.

[Win02] William E Winkler. *Methods for record linkage and bayesian networks*. Tech. rep. Technical report, Statistical Research Division, US Census Bureau . . ., 2002.

[BM03] Mikhail Bilenko and Raymond J Mooney. “Adaptive duplicate detection using learnable string similarity measures”. In: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2003, pp. 39–48.

[FKC03] David Ferraiolo, D Richard Kuhn, and Ramaswamy Chandramouli. *Role-based access control*. Artech House, 2003.

[KAM03a] Anastasios Kementsietsidis, Marcelo Arenas, and Renée J Miller. “Managing data mappings in the hyperion project”. In: *Proceedings 19th International Conference on Data Engineering (Cat. No. 03CH37405)*. IEEE. 2003, pp. 732–734.

[KAM03b] Anastasios Kementsietsidis, Marcelo Arenas, and Renée J Miller. “Mapping data in peer-to-peer systems: Semantics and algorithmic issues”. In: *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. 2003, pp. 325–336.

[KPR03] Jon Kleinberg, Christos Papadimitriou, and Prabhakar Raghavan. "Auditing boolean attributes". In: *Journal of Computer and System Sciences* 66.1 (2003), pp. 244–253.

[Ng+03] Wee Siong Ng et al. "PeerDB: A P2P-based system for distributed data sharing". In: *Proceedings 19th International Conference on Data Engineering (Cat. No. 03CH37405)*. IEEE. 2003, pp. 633–644.

[And+04] Periklis Andritsos et al. "Kanata: adaptation and evolution in data sharing systems". In: *ACM SIGMOD Record* 33.4 (2004), pp. 32–37.

[Cla04] David E Clark. "Practical introduction to record linkage for injury research". In: *Injury Prevention* 10.3 (2004), pp. 186–191.

[Cli+04] Chris Clifton et al. "Privacy-preserving data integration and sharing". In: *Proceedings of the 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*. ACM. 2004, pp. 19–26.

[Fra04] Interoperability Framework. "EUROPEAN INTEROPERABILITY FRAMEWORK FOR PAN-EUROPEAN eGOVERNMENT SERVICES". In: (2004).

[AS04] Mohammad A Al-Kahtani and Ravi Sandhu. "Rule-based RBAC with negative authorization". In: *20th Annual Computer Security Applications Conference*. IEEE. 2004, pp. 405–415.

[KA04] Anastasios Kementsietsidis and Marcelo Arenas. "Data sharing through query translation in autonomous sources". In: *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. 2004, pp. 468–479.

[Riz+04] Shariq Rizvi et al. "Extending query rewriting techniques for fine-grained access control". In: *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. 2004, pp. 551–562.

[WWJ04] Lingyu Wang, Duminda Wijesekera, and Sushil Jajodia. "A logic-based framework for attribute based access control". In: *Proceedings of the 2004 ACM workshop on Formal methods in security engineering*. 2004, pp. 45–55.

[Aro+05] Patricia C. Arocena et al. "Data Sharing in the Hyperion Peer Database System". In: *VLDB*. 2005.

[BS05] Elisa Bertino and Ravi Sandhu. “Database security-concepts, approaches, and challenges”. In: *IEEE Transactions on Dependable and secure computing* 2.1 (2005), pp. 2–19.

[CGM05] Surajit Chaudhuri, Venkatesh Ganti, and Rajeev Motwani. “Robust identification of fuzzy duplicates”. In: *21st International Conference on Data Engineering (ICDE'05)*. IEEE. 2005, pp. 865–876.

[Fag+05] Ronald Fagin et al. “Data exchange: semantics and query answering”. In: *Theoretical Computer Science* 336.1 (2005), pp. 89–124.

[Jos+05] J.B.D. Joshi et al. “A generalized temporal role-based access control model”. In: *IEEE Transactions on Knowledge and Data Engineering* 17.1 (2005), pp. 4–23. DOI: 10.1109/TKDE.2005.1.

[Kol05] Phokion G Kolaitis. “Schema mappings, data exchange, and metadata management”. In: *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 2005, pp. 61–75.

[YT05] Eric Yuan and Jin Tong. “Attributed based access control (ABAC) for web services”. In: *IEEE International Conference on Web Services (ICWS'05)*. IEEE. 2005.

[Fan+06] Wenfei Fan et al. “A view based security framework for XML”. In: (2006).

[FAL06] Keith Frikken, Mikhail Atallah, and Jiangtao Li. “Attribute-based access control with hidden policies and hidden credentials”. In: *IEEE Transactions on Computers* 55.10 (2006), pp. 1259–1270.

[GB06] Lifang Gu and Rohan Baxter. “Decision models for record linkage”. In: *Data mining*. Springer. 2006, pp. 146–160.

[Min06] What Is Data Mining. “Data mining: Concepts and techniques”. In: *Morgan Kaufinann* 10 (2006), pp. 559–569.

[BAK07] Nick Bakis, Ghassan Aouad, and Mike Kagioglou. “Towards distributed product data sharing environments—progress so far and future challenges”. In: *Automation in Construction* 16.5 (2007), pp. 586–595.

[BG07] Indrajit Bhattacharya and Lise Getoor. “Collective entity resolution in relational data”. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1.1 (2007), 5–es.

[Cha+07a] David W Chadwick et al. “Multi-session separation of duties (MSoD) for RBAC”. In: *2007 IEEE 23rd International Conference on Data Engineering Workshop*. IEEE. 2007, pp. 744–753.

[Cha+07b] Surajit Chaudhuri et al. “Example-driven design of efficient record matching queries.” In: *VLDB*. Vol. 7. 2007, pp. 327–338.

[CG07] Peter Christen and Karl Goiser. “Quality and complexity measures for data linkage and deduplication”. In: *Quality measures in data mining*. Springer, 2007, pp. 127–151.

[DS07] Nilesh Dalvi and Dan Suciu. “Efficient query evaluation on probabilistic databases”. In: *The VLDB Journal* 16.4 (2007), pp. 523–544.

[HSW07] Thomas N Herzog, Fritz J Scheuren, and William E Winkler. *Data quality and record linkage techniques*. Springer Science & Business Media, 2007.

[LCW07] Luis Leitao, Pável Calado, and Melanie Weis. “Structure-based inference of XML similarity for fuzzy duplicate detection”. In: *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. 2007, pp. 293–302.

[MS07] Gerome Miklau and Dan Suciu. “A formal analysis of information disclosure in data exchange”. In: *Journal of Computer and System Sciences* 73.3 (2007), pp. 507–534.

[ND07] Alan Nash and Alin Deutsch. “Privacy in GLAV information integration”. In: *International Conference on Database Theory*. Springer. 2007, pp. 89–103.

[RSH07] Vibhor Rastogi, Dan Suciu, and Sungho Hong. “The boundary between privacy and utility in data publishing”. In: *Proceedings of the 33rd international conference on Very large data bases*. Citeseer. 2007, pp. 531–542.

[Sca+07] Monica Scannapieco et al. “Privacy preserving schema and data matching”. In: *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. ACM. 2007, pp. 653–664.

[Sna07] Chakkrit Snae. “A comparison and analysis of name matching algorithms”. In: *International Journal of Applied Science. Engineering and Technology* 4.1 (2007), pp. 252–257.

[Chr08] Peter Christen. “Automatic record linkage using seeded nearest neighbour and support vector machine classification”. In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2008, pp. 151–159.

[Dwo08] Cynthia Dwork. “Differential privacy: A survey of results”. In: *International conference on theory and applications of models of computation*. Springer. 2008, pp. 1–19.

[Lin+08] Jiayuan Lin et al. “View-based Access Control Mechanism for Spatial Database”. In: (Dec. 2008). DOI: 10.1117/12.815569.

[AK09] Arvind Arasu and Raghav Kaushik. “A grammar-based entity representation framework for data cleaning”. In: *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*. 2009, pp. 233–244.

[Ben+09] Omar Benjelloun et al. “Swoosh: a generic approach to entity resolution”. In: *The VLDB Journal* 18.1 (2009), pp. 255–276.

[Cho+09] Eun-Sun Cho et al. “Fine-Grained View-Based Access Control for RDF Cloaking”. In: *2009 Ninth IEEE International Conference on Computer and Information Technology*. Vol. 1. 2009, pp. 336–341. DOI: 10.1109/CIT.2009.102.

[Cir+09] Valentina Ciriani et al. “Keep a few: Outsourcing data while maintaining confidentiality”. In: *European Symposium on Research in Computer Security*. Springer. 2009, pp. 440–455.

[Fag+09] Ronald Fagin et al. “Clio: Schema mapping creation and data exchange”. In: *Conceptual modeling: foundations and applications*. Springer, 2009, pp. 198–236.

[Fis+09] Jeffrey Fischer et al. “Fine-grained access control with object-sensitive roles”. In: *European Conference on Object-Oriented Programming*. Springer. 2009, pp. 173–194.

[Guo+09] Honglei Guo et al. “Address standardization with latent semantic association”. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2009, pp. 1155–1164.

[HM09] Oktie Hassanzadeh and Renee J Miller. “Creating probabilistic databases from duplicated data”. In: *The VLDB Journal* 18.5 (2009), pp. 1141–1166.

[TK09] Balder Ten Cate and Phokion G Kolaitis. “Structural characterizations of schema-mapping languages”. In: *Proceedings of the 12th International Conference on Database Theory*. 2009, pp. 63–72.

[VMI09] Millist W Vincent, Mukesh Mohania, and Mizuho Iwaihara. “Detecting privacy violations in database publishing using disjoint queries”. In: *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*. ACM. 2009, pp. 252–262.

[AGK10] Arvind Arasu, Michaela Götz, and Raghav Kaushik. “On active learning of record matching packages”. In: *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. 2010, pp. 783–794.

[Elm+10] Hazem Elmeleegy et al. “Preserving privacy and fairness in peer-to-peer data integration”. In: *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. 2010, pp. 759–770.

[Fer10] Elena Ferrari. “Access control in data management systems”. In: *Synthesis lectures on data management* 2.1 (2010), pp. 1–117.

[GL10] Alban Gabillon and Léo Letouzey. “A view based access control model for SPARQL”. In: *2010 Fourth International Conference on Network and System Security*. IEEE. 2010, pp. 105–112.

[HN10] Junbeom Hur and Dong Kun Noh. “Attribute-based access control with efficient revocation in data outsourcing systems”. In: *IEEE Transactions on Parallel and Distributed Systems* 22.7 (2010), pp. 1214–1221.

[Ina+10] Ali Inan et al. “Private record matching using differential privacy”. In: *Proceedings of the 13th International Conference on Extending Database Technology*. ACM. 2010, pp. 123–134.

[KR10] Hanna Köpcke and Erhard Rahm. “Frameworks for entity matching: A comparison”. In: *Data & Knowledge Engineering* 69.2 (2010), pp. 197–210.

[NH10] Felix Naumann and Melanie Herschel. “An introduction to duplicate detection”. In: *Synthesis Lectures on Data Management* 2.1 (2010), pp. 1–87.

[OZ10] Stanley RM Oliveira and Osmar R Zaiane. “Privacy preserving clustering by data transformation”. In: *Journal of Information and Data Management* 1.1 (2010), pp. 37–37.

[She+10] Weiming Shen et al. “Systems integration and collaboration in architecture, engineering, construction, and facilities management: A review”. In: *Advanced engineering informatics* 24.2 (2010), pp. 196–207.

[WYP10] William E Winkler, Willian Yancey, and EH Porter. “Fast record linkage of very large files in support of decennial and administrative records projects”. In: *Proceedings of the Section on Survey Research Methods, American Statistical Association*. 2010, pp. 2120–30.

[Ale+11] Bogdan Alexe et al. “Characterizing schema mappings via data examples”. In: *ACM Transactions on Database Systems (TODS)* 36.4 (2011), pp. 1–48.

[BM11] Jason Bau and John C Mitchell. “Security modeling and analysis”. In: *IEEE Security & Privacy* 9.3 (2011), pp. 18–25.

[Ber+11] Sonia Bergamaschi et al. “Data integration”. In: *Handbook of Conceptual Modeling*. Springer, 2011, pp. 441–476.

[BMR11] Philip A Bernstein, Jayant Madhavan, and Erhard Rahm. "Generic schema matching, ten years later". In: *Proceedings of the VLDB Endowment* 4.11 (2011), pp. 695–701.

[Chr11] Peter Christen. "A survey of indexing techniques for scalable record linkage and deduplication". In: *IEEE transactions on knowledge and data engineering* 24.9 (2011), pp. 1537–1555.

[EEL11] Hazem Elmeleegy, Ahmed Elmagarmid, and Jaewoo Lee. "Leveraging query logs for schema mapping generation in U-MAP". In: *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. 2011, pp. 121–132.

[HPK11] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[KR11] Raghav Kaushik and Ravi Ramamurthy. "Efficient Auditing for Complex SQL Queries". In: *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*. SIGMOD '11. 2011, pp. 697–708.

[LF11] Jie Liu and Wenfei Fan. "Polymorphic queries for P2P systems". In: *Information Systems* 36.5 (2011), pp. 825–842.

[RDG11] Vibhor Rastogi, Nilesh Dalvi, and Minos Garofalakis. "Large-scale collective entity matching". In: *arXiv preprint arXiv:1103.2410* (2011).

[Wan+11a] Jiannan Wang et al. "Entity Matching: How Similar is Similar". In: *Proc. VLDB Endow.* (2011), pp. 622–633.

[Wan+11b] Jiannan Wang et al. "Entity matching: How similar is similar". In: *Proceedings of the VLDB Endowment* 4.10 (2011), pp. 622–633.

[Chr12] Peter Christen. *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer Science & Business Media, 2012.

[DHI12] AnHai Doan, Alon Halevy, and Zachary Ives. *Principles of data integration*. Elsevier, 2012.

[HFB12] Laurinda B Harman, Cathy A Flite, and Kesa Bond. "Electronic health records: privacy, confidentiality, and security". In: *AMA Journal of Ethics* 14.9 (2012), pp. 712–719.

[Hua+12] Jingwei Huang et al. “A framework integrating attribute-based policies into role-based access control”. In: *Proceedings of the 17th ACM symposium on Access Control Models and Technologies*. 2012, pp. 187–196.

[Jun+12] Youna Jung et al. “A survey of security issue in multi-agent systems”. In: *Artificial Intelligence Review* 37.3 (2012), pp. 239–260.

[LJ12] Alexandros Labrinidis and Hosagrahar V Jagadish. “Challenges and opportunities with big data”. In: *Proceedings of the VLDB Endowment* 5.12 (2012), pp. 2032–2033.

[MPB12] Néjib Moalla, Hervé Panetto, and Xavier Boucher. “Interopérabilité et partage de connaissances”. In: *Revue des Sciences et Technologies de l’Information-Série ISI: Ingénierie des Systèmes d’Information* 17.4 (2012), pp. 7–16.

[Bel+13] Khalid Belhajjame et al. “Incrementally improving dataspaces based on user feedback”. In: *Information Systems* 38.5 (2013), pp. 656–687.

[VC13] Dinusha Vatsalan and Peter Christen. “Sorted nearest neighborhood clustering for efficient private blocking”. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2013, pp. 341–352.

[And14] Jason Andress. *The basics of information security: understanding the fundamentals of InfoSec in theory and practice*. Syngress, 2014.

[Aya+14] Vanessa Ayala-Rivera et al. “A Systematic Comparison and Evaluation of k-Anonymization Algorithms for Practitioners”. In: *Transactions on Data Privacy* 7 (Dec. 2014), pp. 337–370.

[Don+14] Xin Dong et al. “Achieving an effective, scalable and privacy-preserving data sharing service in cloud computing”. In: *Computers & security* (2014), pp. 151–164.

[Had+14] Mehdi Haddad et al. “Access control for data integration in presence of data dependencies”. In: *International Conference on Database Systems for Advanced Applications*. Springer. 2014, pp. 203–217.

[Lia+14] Huizhi Liang et al. “Noise-tolerant approximate blocking for dynamic real-time entity resolution”. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2014, pp. 449–460.

[Ste+14] Rebecca C Steorts et al. “A comparison of blocking methods for record linkage”. In: *International conference on privacy in statistical databases*. Springer. 2014, pp. 253–268.

[Wu+14] Xindong Wu et al. “Data mining with big data”. In: *IEEE transactions on knowledge and data engineering* 26.1 (2014), pp. 97–107.

[BBM15] Sabine Brunswicker, Elisa Bertino, and Sorin Matei. “Big data for open digital innovation–a research roadmap”. In: *Big Data Research* 2.2 (2015), pp. 53–58.

[Don+15] Xinhua Dong et al. “Secure sensitive data sharing on a big data platform”. In: *Tsinghua science and technology* 20.1 (2015), pp. 72–80.

[Fan+15] Hua Fang et al. “A survey of big data research”. In: *IEEE network* 29.5 (2015), pp. 6–9.

[Hu+15] Vincent C Hu et al. “Attribute-based access control”. In: *Computer* 48.2 (2015), pp. 85–88.

[Pan+15] Xiaoman Pan et al. “Unsupervised entity linking with abstract meaning representation”. In: *Proceedings of the 2015 conference of the north american chapter of the association for computational linguistics: Human language technologies*. 2015, pp. 1130–1139.

[RJK15] Qasim Mahmood Rajpoot, Christian Damsgaard Jensen, and Ram Krishnan. “Attributes enhanced role-based access control model”. In: *International Conference on Trust and Privacy in Digital Business*. Springer. 2015, pp. 3–17.

[KW16] Harlan M Krumholz and Joanne Waldstreicher. “The Yale Open Data Access (YODA) project–a mechanism for data sharing”. In: *The New England journal of medicine* 375.5 (2016), pp. 403–405.

[RM16] Fatemeh Rezaeibagha and Yi Mu. “Distributed clinical data sharing via dynamic access-control policy transformation”. In: *International journal of medical informatics* (2016), pp. 25–31.

[DLK17] Edward S Dove, Graeme T Laurie, and Bartha M Knoppers. "Data sharing and privacy". In: *Genomic and Precision Medicine*. 2017, pp. 143–160.

[God17] Michelle Goddard. "The EU General Data Protection Regulation (GDPR): European regulation that has a global impact". In: *International Journal of Market Research* 59.6 (2017), pp. 703–705.

[GMB17] Marco Guarnieri, Srdjan Marinovic, and David Basin. "Securing databases from probabilistic inference". In: *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE. 2017, pp. 343–359.

[KMG+17] Angelika Kimmig, Alex Memory, Lise Getoor, et al. "A collective, probabilistic approach to schema mapping". In: *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. IEEE. 2017, pp. 921–932.

[Pan+17] Fatemah Panahi et al. "Towards Interactive Debugging of Rule-based Entity Matching." In: *EDBT*. 2017, pp. 354–365.

[RMV17] P Muthi Reddy, SH Manjula, and KR Venugopal. "Secure data sharing in cloud computing: a comprehensive review". In: *International Journal of Computer (IJC)* 25.1 (2017), pp. 80–115.

[Sin+17] Rohit Singh et al. "Synthesizing entity matching rules by examples". In: *Proceedings of the VLDB Endowment* 11.2 (2017), pp. 189–202.

[TPN17] Fabian Tschirschnitz, Thorsten Papenbrock, and Felix Naumann. "Detecting inclusion dependencies on very many tables". In: *ACM Transactions on Database Systems (TODS)* 42.3 (2017), pp. 1–29.

[Vat+17] Dinusha Vatsalan et al. "Privacy-preserving record linkage for big data: Current approaches and research challenges". In: *Handbook of Big Data Technologies*. Springer, 2017, pp. 851–895.

[Cos+18] Gianpiero Costantino et al. "Privacy preserving distributed computation of private attributes for collaborative privacy aware usage control systems". In: *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE. 2018, pp. 315–320.

[De 18] Paul B De Laat. “Algorithmic decision-making based on machine learning from Big Data: Can transparency restore accountability?” In: *Philosophy & technology* 31.4 (2018), pp. 525–541.

[Dev+18] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).

[FSR18] Martin Franke, Ziad Sehili, and Erhard Rahm. “Parallel Privacy-preserving Record Linkage using LSH-based Blocking.” In: *IoTBDS*. 2018, pp. 195–203.

[Kol18] Phokion G Kolaitis. “Reflections on schema mappings, data exchange, and metadata management”. In: *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. 2018, pp. 107–109.

[Li+18] Jin Li et al. “Secure attribute-based data sharing for resource-limited users in cloud computing”. In: *Computers & Security* 72 (2018), pp. 1–12.

[PP18] George Papadakis and Themis Palpanas. “Web-scale, schema-agnostic, end-to-end entity resolution”. In: *The Web Conference (WWW)*. 2018.

[Sim+18] Giovanni Simonini et al. “Schema-agnostic progressive entity resolution”. In: *IEEE Transactions on Knowledge and Data Engineering* 31.6 (2018), pp. 1208–1221.

[Vim+18] Sabrina De Capitani di Vimercati et al. “Confidentiality protection in large databases”. In: *A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years*. Springer, 2018, pp. 457–472.

[AH19] Juba Agoun and Mohand-Saıd Hacid. “Data sharing in presence of access control policies”. In: *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer. 2019, pp. 301–309.

[Ben+19] Domenico Beneventano et al. “Entity resolution and data fusion: An integrated approach”. In: *SEBD 2019: 27th Italian Symposium on Advanced Database Systems*. Vol. 2400. CEUR-WS. org. 2019.

[Chr+19] Vassilis Christophides et al. “End-to-end entity resolution for big data: A survey”. In: *arXiv preprint arXiv:1905.06397* (2019).

[Eft+19] Vasilis Efthymiou et al. “MinoanER: Schema-agnostic, non-iterative, massively parallel resolution of web entities”. In: *arXiv preprint arXiv:1905.06170* (2019).

[Qui+19] Martha Quinn et al. “Electronic health records, communication, and data sharing: challenges and opportunities for improving the diagnostic process”. In: *Diagnosis* 6.3 (2019), pp. 241–248.

[Sha+19] Richard Shay et al. “Don’t Even Ask: Database Access Control through Query Control”. In: *ACM SIGMOD Record* 47.3 (2019), pp. 17–22.

[AH20] Juba Agoun and Mohand-Saıd Hacid. “Data Publishing: Availability of Data Under Security Policies”. In: *International Symposium on Methodologies for Intelligent Systems*. Springer. 2020, pp. 277–286.

[HKP20] Yuncong Hu, Sam Kumar, and Raluca Ada Popa. “Ghostor: Toward a secure data-sharing system from decentralized trust”. In: *17th Symposium on Networked Systems Design and Implementation*. 2020, pp. 851–877.

[LPX20] Xiuqing Lu, Zhenkuan Pan, and Hequn Xian. “An efficient and secure data sharing scheme for mobile devices in cloud computing”. In: *Journal of Cloud Computing* 9.1 (2020), pp. 1–13.

[PIP20] George Papadakis, Ekaterini Ioannou, and Themis Palpanas. “Entity Resolution: Past, Present and Yet-to-Come.” In: *EDBT*. 2020, pp. 647–650.

[Pat+20] Abdullah Al-Noman Patwary et al. “Authentication, Access Control, Privacy, Threats and Trust Management Towards Securing Fog Computing Environments: A Review”. In: *arXiv preprint arXiv:2003.00395* (2020).

[TSS20] Kai-Sheng Teong, Lay-Ki Soon, and Tin Tin Su. “Schema-Agnostic Entity Matching using Pre-trained Language Models”. In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2020, pp. 2241–2244.

[EAH21] Peter Edemekong, Pavan Annamaraju, and Micelle Haydel. “Health insurance portability and accountability act”. In: *StatPearls* (2021).

[AH22] Juba Agoun and Mohand-Saıd Hacid. “Access control based on entity matching for secure data sharing”. In: *Serv. Oriented Comput. Appl.* 16.1 (2022), pp. 31–44. DOI: 10.1007/s11761-021-00331-3. URL: https://doi.org/10.1007/s11761-021-00331-3.