UNIVERSITY OF SOUTHAMPTON

FACULTY OF PHYSICAL AND APPLIED SCIENCES

Electronics and Computer Science

Secure Data Integration Systems

by

Fatmah Y. Akeel

Thesis for the degree of Doctor of Philosophy

October 2017

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL AND APPLIED SCIENCES
Electronics and Computer Science

Doctor of Philosophy

SECURE DATA INTEGRATION SYSTEMS

by Fatmah Y. Akeel

As the web moves increasingly towards publishing data, a significant challenge arises when integrating data from diverse sources that have heterogeneous security and privacy policies and requirements. Data Integration Systems (DIS) are concerned with integrating data from multiple data sources to resolve users' queries. DIS are prone to data leakage threats, e.g. unauthorised disclosure or secondary use of the data, that compromise the data's confidentiality and privacy. We claim that these threats are caused by the failure to implement or correctly employ confidentiality and privacy techniques, and by the failure to consider the trust levels of system entities, from the very start of system development. Data leakage also results from a failure to capture or implement the security policies imposed by the data providers on the collection, processing, and disclosure of personal and sensitive data.

This research proposes a novel framework, called SecureDIS, to mitigate data leakage threats in DIS. Unlike existing approaches that secure such systems, SecureDIS helps software engineers to lessen data leakage threats during the early phases of DIS development. It comprises six components that represent a conceptualised DIS architecture: data and data sources, security policies, integration approach, integration location, data consumers, and System Security Management (SSM). Each component contains a set of informal guidelines written in natural language to be used by software engineers who build and design a DIS that handles sensitive and personal data.

SecureDIS has undergone two rounds of review by experts to confirm its validity, resulting in the guidelines being evaluated and extended. Two approaches were adopted to ensure that SecureDIS is suitable for software engineers. The first was to formalise the guidelines by modelling a DIS with the SecureDIS security policies using Event-B formal methods. This verified the correctness and consistency of the model. The second approach assessed SecureDIS's applicability to a real data integration project by using a case study. The case study addressed the experts' concerns regarding the ability to apply the proposed guidelines in practice.

Based on the research methods adopted, SecureDIS was accepted by software engineers following their reviews. In addition, the SecureDIS guidelines were all applicable to the DIS investigated by the case study.

Contents

List of Figures

List of Tables

Declaration of Authorship

I, Fatmah Y. Akeel , declare that the thesis entitled *Secure Data Integration Systems* and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as: (Akeel et al., 2013),(Akeel et al., 2014),(Akeel et al., 2015) and (Akeel et al., 2016)

Signed:.....

Date:.....

Acknowledgements

To all passionate scholars who find joy in pushing the limits.

To my mother Dr. Fawziah Alabdulaly

I am glad that Allah almighty has granted me the wisdom, strength, and good health to complete this thesis. I am sincerely grateful for many individuals who crossed my path during my studies, who provided me with their support and care.

I am grateful for my supervisors Dr. Gary Wills, Dr. Andrew Gravell, and Dr. Federica Paci for their continuous support and the impart of their knowledge and expertise in this thesis. I also would like to convey my appreciation for Dr. Asieh Salehi for her kind support.

My thanks are also extended to Mr. Mohammed Alshahri, Mr. Wael Alalwani, Mr. Muhammed Saleem, Dr. Jalal Alowaibdi and Dr. Theeb Alqahtani for their help and support during the early stages of the PhD.

And of course I am extremely grateful for my family, who were behind me every step of the way, my mother Dr. Fawzaiah and my father Yousef, my sisters Batoul and Bothinah, my brothers Abdulrahman, Abdullah, Omar and Ahmad, and my husband Khalid Alashgar. I am also thankful to my lovely children Rand, Nasser, and Leen who were patient with me through this journey.

Finally, I would like to acknowledge my friends who supported me greatly with their kindness and care, Dr. Areeb Alowaisheg, Ms. Dalal Alazizy, Dr. Norah Alrajebah, Dr. Alaa Mashat, Dr. Norah Almuhanna, Dr. Norah Alothaman, Mr. Nawfal Fadhel, Ms. Shorouq Alansari, and Dr. Shre Chatterjee.

Chapter 1

Introduction

The web nowadays is witnessing an immense shift towards data publishing. Data is being described as big, open, and linked, and is offered in many different formats. Data integration is a useful approach that can be utilised to obtain a unified view of the data coming from multiple, heterogeneous, data sources (Lenzerini, 2002; Calì et al., 2006). Data is integrated for sharing and collaboration, by eliminating the need to query each data source separately when collecting information (Clifton et al., 2004). The unified view produces useful information for data analysis to enable it to find patterns and abnormalities that assist in deriving conclusions and making decisions. As an example, in disaster situations, where human lives are at stake, data integration can help contain problems and find solutions. During 2008 Hurricane Katrina in the USA, integration and collaboration between government agencies was used to contain the problem and help those affected.

The data integration research community emerged over 20 years ago (Chawathe et al., 1994; Batini et al., 1986; Ahmed et al., 1991). The community mainly focuses on issues related to integrating heterogeneous data sources by converting between data models in different contexts. Currently, data integration has gained more interest as many technologies adopt its concepts. For example, traditional data integration has shifted towards Big Data Integration (BDI), where the volume, velocity, variety, and veracity of the data are changed (Dong and Srivastava, 2015). Likewise in the utilisation of the cloud as a platform to integrate data in specific domains, such as in integrating and analysing healthcare information coming from diverse geographical locations (Bahga and Madisetti, 2015); or using the cloud to provide services to other entities, namely integration Platform as a Service (iPaaS) (Palanimalai and Paramasivam, 2015). Regardless of what data integration is currently called and which platform it uses, it is an essential technique that can always accommodate changes in technology and fulfil the needs of applications that rely on data analysis.

Data Integration Systems (DIS) are systems built on the basis of consumers querying several heterogeneous data sources (Dicelie et al., 2001; Gusmini and Leida, 2011; Guo and Fang, 2011). These systems utilise a mediator (a middle layer) to achieve the integration. Data sources used in these systems may include personal or sensitive data; consequently, data integration poses different security and privacy challenges. For example, organisations that provide data for integration have specific expectations as to how that data is collected, processed, disclosed, and protected. These expectations take the form of security and privacy requirements and policies. Failing to consider these requirements imposes threats to the DIS that vary from unauthorised access and secondary use of data, to violating data protection regulations.

Within information security, a generic category of threat called data leakage is concerned with the exposure of sensitive data intentionally or unintentionally (CWE, 2013). We argue that DIS are prone to data leakage threats in several ways. One way is by violating the *confidentiality* of the data provided by the data sources. For example, the queries executed by the mediator expose data to consumers that were not allowed to access that data according to the security policies of the data sources. Another aspect is by violating the *privacy* of the data when a query discloses the data to a consumer that has a purpose different from the one for which they were originally collected by the data source. However, data leakage threats can still materialise in a DIS even when the mediator enforces security policies on the execution of a query that discloses data, only if the consumer is authorised and only if the purpose of the query matches the purpose for which the data was collected. A further aspect is the lack of *trust* in system's entities. For example, it is possible that authorised data consumers, who access sensitive data, do not process the data according to the security policies of the data sources. Consumers may share the data with unauthorised parties or use the data for fraudulent purposes.

The existing literature shows that the data integration community is interested in enhancing approaches to preserving the data's privacy during integration (Clifton et al., 2004; Bhowmick et al., 2006). However, these approaches mainly focus on privacy issues separately and do not consider the combination of Confidentiality, Privacy, and Trust (CPT) properties. Current approaches focus on one property or a combination of two, trust usually being assumed in the system's entities. To overcome data leakage in any DIS, we argue that it is important to adopt the concept of security by design (McGraw, 2004), to enable secure systems to be built *ab initio*. The novelty of our approach to mitigate data leakage comes from the ability to combine the perspectives of software engineering, information security and data integration, which is unique within the data integration community.

Several tactics are introduced in this research to support software engineers in the development of secure DIS. *The first tactic* is to conduct a threat analysis to elicit data leakage threats occurring in DIS. The threats elicited can inform development thus ensuring secure design. *The second tactic* is a novel architectural framework, Secure Data

Integration Systems (SecureDIS), that assists software engineers in designing DIS to be secure from the start. Each component of SecureDIS has a list of guidelines designed to mitigate data leakage threats. *The third tactic* is a formalisation of DIS security policies. To demonstrate the ability to implement SecureDIS, the informal SecureDIS guidelines are formalised using Event-B formal methods. *The final tactic* is a demonstration that SecureDIS is applicable in reality. A case study is conducted to assess the applicability of SecureDIS to a real data integration project.

1.1 Research Problem

The research problem is concerned with addressing a generic security threat called data leakage, it can be caused by any of the following:

- The lack of capture, implementation, or maintenance of the security and privacy requirements of the data sources and the entities involved in the integration process. The integration process usually addresses the data level of the integration, which is concerned with the actual combination of the data to resolve queries, and gives little or no attention to the security and privacy policies associated with the data sources. Failing to address these policies within the integration process is a security violation. This can occur by completely overlooking the policies or by the lack of integrating the policies and using the results to influence the integration process. This consequently leads to the lack of maintaining the policies throughout the complete integration process (see Chapter 4).

- The lack of effectively using or choosing security techniques leads to violating security or privacy properties. For example, the weakness, misconfiguration, or inappropriate choice of access controls models (Braghin et al., 2003; Watson, 2007; Pistoia et al., 2007) causes data leakage.

- The lack of depth in understanding security or privacy properties. For example, not considering the nature of the data, whether it is Personally Identifiable Information, or Quasi-Identifiers[1], leads to a wide spectrum of inference attacks occurring (Fung et al., 2012; Boyens et al., 2004; Whang and Garcia-Molina, 2012; Clifton et al., 2004).

- The use of external entities to assist in the integration process. For example, using cloud services or third-party entities exacerbates security threats as these may not protect the data according to its security policies or may allow transitive trust in handling the data to unauthorised entities (Fung et al., 2012).

[1] Quasi-Identifiers (QIDs): a set of attributes that could potentially identify an individual (Mohammed et al., 2011)

- The mere nature of the distributed DIS in which data travels from its sources through several layers to reach the consumers. Hence, the data and its policies may suffer exposure at any layer if security measures are not in place. Manan et al. (2011) discusses security concerns in a general distributed environment, but applying the same concerns to a DIS context, the following concerns arise:
  - What privacy policy is used by the applications accessing the integrated data?
  - Is the server used for data integration secure and trusted?
  - Is the trusted third party assisting in the integration complying with corporate or legislative policies?
- The violations from the *data consumers*' side are possible when the system assumes it can trust the consumers. Not considering the risks associated with data consumers in handling sensitive data can result in secondary use of data (Paci et al., 2013). Once the data is disclosed to data consumers, they could misuse the data by sharing it with unauthorised parties, or use it for purposes other than the data provider's intended purposes.

1.2 Research Scope

We argue that employing the concept of security by design will mitigate the threats from the start. The purpose is to design a DIS that achieves collaboration and data sharing, while maintaining the security policies of the participating entities. Through maintaining confidentiality and by protecting privacy while considering trust earlier in the development, we claim that data leakage threats can be mitigated. The scope of this research covers:

1. The data leakage threats occurring in DIS with middle layers.
2. The combination of the CPT properties, so that the threats and proposed mitigation against them may be studied.
3. The DIS that combine data from different organisations. These organisations have different security and privacy requirements.
4. The main DIS components that represent DIS with middle layers. Having a holistic perspective of DIS when investigating security and privacy.
5. The software engineering perspective in designing the mitigation approach, creating security policies from it, and applying it to a real data integration project.

1.3 Research Aim and Questions

The aim of this research is to develop an approach to mitigating data leakage threats in those DIS with middle layers. The approach targets software engineers and focuses on the CPT properties. We argue that the combination of the CPT properties should be closely related to the proposed approach. By adopting the concept of security by design and considering the main components of the DIS, a system would be built to be secure from the beginning.

As far as is known, no literature has addressed the combination of CPT in a DIS context to mitigate data leakage, which encourages this research. The research questions of this thesis are:

RQ1: What data leakage threats affect personal and sensitive data in DIS, focusing on the CPT properties?

RQ2: How can software engineers mitigate data leakage threats during the design of DIS?

RQ3: To what extent is the proposed mitigation approach against data leakage applicable to a real data integration project?

1.4 Contributions

This research combines three different research areas. It targets data integration as the main context, explores information security within the integration, and uses secure software engineering as an approach to mitigate security threats. This research is therefore able to provide both breadth by combining the fields, and depth by focusing on an implementable approach. Security threats will always arise in information systems as attacks are getting more advanced. However, encouraging the building of secure systems by design accommodates security earlier in the development and allows further security techniques to be employed.

This work contains the following contributions useful to the research community:

1. Data leakage threat analysis on DIS guided by a novel perspective of the combination of CPT properties that was also used to propose the mitigation approach (Akeel et al., 2014). The analysis is property-driven; hence, it can be useful to the information security academic domain.

2. SecureDIS, a novel framework to mitigate data leakage threats containing the architectural components of a DIS with a middle layer (Akeel et al., 2015). The

framework can be useful to the information security and software engineering academic domains as it provides a real system abstraction that can be used for further research.

3. A set of guidelines, as part of SecureDIS, assisting software engineers in building secure DIS by design (Akeel et al., 2013). The guidelines can be used in software industry as they target the system designers.

4. The formalisation of the SecureDIS guidelines using Event-B formal method to create DIS security policies that demonstrates the implementability of the guidelines by software engineers (Akeel et al., 2016). The formalisation is useful to the software industry as it achieves accuracy and consistency in the design. Also, it is useful in information security academic domain as it provides a use case of building security policies.

5. The application of SecureDIS and its guidelines to a real data integration project through a case study. The ability to conduct security such case studies where information is confidential is a valuable aspect in the domain of information security. The case study serves both the software industry and the information security domains by bridging the gap between theory and practice.

1.5 Thesis Structure

Chapter 2 provides a background on the DIS and architecture, as well as a background of software security and the activities to incorporate security into development. The issue of data leakage is discussed in the context of DIS. The chapter also explains related work that aims at securing DIS. The final part of the chapter provides a literature critique highlighting the gaps identified.

In Chapter 3, the research questions are presented. In addition, the research methods employed to answer the research are discussed. The chapter concludes with a description of the research methodology adopted throughout this work.

To begin the process of understanding the research problem, Chapter 4 explains the process in several steps. The first step describes the architecture of a DIS with a middle layer and reports on qualitative expert reviews that aim to confirm the architectural components and the data leakage locations. The remaining steps introduce a threat analysis to elicit data leakage threats in the architecture. This chapter helps in constructing the components of the SecureDIS framework which is presented in the next chapter.

The Secure Data Integration System (SecureDIS) framework is presented in Chapter 5, which is the main contribution of this study to mitigate data leakage. The chapter

discusses the components of SecureDIS and the proposed guidelines. To confirm the validity of the proposed guidelines, the chapter also explains the details of the validation of SecureDIS by a second set of expert reviews that include quantitative and qualitative analysis of the results and introduces changes to the SecureDIS framework.

In an attempt to use SecureDIS in practice, the framework and its guidelines were used to model the security policies governing the core interaction between consumers and data sources. The policies were modelled using the Event-B formal method. Chapter 6 discusses the details of this activity.

To address the experts' suggestions in assessing the applicability of the SecureDIS, a case study was conducted on an actual DIS project as presented in Chapter 7. The case study discusses the extent to which the SecureDIS guidelines were applicable to the DIS project.

Chapter 8 summarises the main points of this work and presents the future work.

Chapter 2

Literature Review

This research focuses on the security aspects of data integration that are essential to guarantee that the process continues to provide, in a secure fashion, the functionality needed. The research problem, discussed in Section 1.1, summarises the threats of data leakage in DIS. The aim of this study is to devise an approach to mitigate those threats during system development by following the concept of security by design. This chapter reviews the literature relevant to the research problem and its proposed solution. To build an understanding of the fundamental elements of this study, the literature covers three main research areas: data integration, security (including privacy and trust), and software engineering. Figure 2.1 illustrates the overlap between these areas and highlights the research problem as label *a*, and the proposed solution as label *b*.

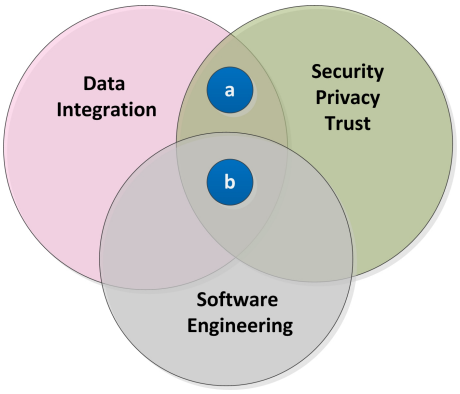

Figure 2.1: Research Areas Covered in the Literature Review

In this chapter, Section 2.1 provides a background of the DIS to help understand the context of the research problem. Section 2.2 discusses the causes of data leakage in DIS as a security threat with an overview of security, privacy, and trust. Section 2.3 discusses the approaches to mitigate data leakage in DIS while maintaining security. Section 2.4 presents the concept of security by design and discusses the available tools to achieve it. Section 2.5 critiques the literature and emphasises the research gap. The final section summarises the main points and introduces the next chapter.

2.1 Data Integration Systems (DIS) Background

This section discusses the definition of data integration, integration approaches, DIS architectures, and data integration applications.

2.1.1 Data Integration

Data, as defined by the Open Data community, is the "*qualitative or quantitative statements or numbers that are assumed to be factual, and not the product of analysis or interpretation*" (Maude, 2012). Data occurs on the web in different formats: structured, such as database relations, unstructured, such as text and geospatial data, or semi-structured, such as XML[1], RDF[2], and OWL[3] (Harris et al., 2007).

With hardware's ability to store significant datasets, the concept of data has expanded to accommodate different types of data, such as open data, big data and linked data. *Open Data* refers to datasets accessible through the internet in a digital format without any restrictions on use or redistribution (Maude, 2012). *Big Data* refers to data with a significant scale, for which traditional database management tools cannot be used to capture, store, manage, or analyse it (Yiu, 2012). *Linked Data* is the result of linking different data on the web using semantic links, such as RDFs (Bizer et al., 2009).

Data integration is defined as the process of combining data from multiple sources to provide the user with a unified result from the combination (Lenzerini, 2002; Calì et al., 2006; Huang et al., 2008). It can also be defined as transferring data from a source to a destination, such as transferring data from old databases to new ones (Doan and Halevy, 2005). Integration is the process of combination; it can be integrating data, services, processes, or features from multiple sources.

Data Integration has been studied for almost 30 years (see Batini et al. (1986)), and is still an area of current research. Traditional data integration has been studied by many research communities (Harris et al., 2007), such as those specialising in databases (Doan and Halevy, 2005), data mining (Calì et al., 2006), artificial intelligence (Noy, 2004), and ontology (Wache et al., 2001; Noy, 2004).

As volumes of data are growing significantly, smart organisations are now improving how the business is done rather than simply doing it, which is achieved by data integration (Loshin, 2010). Integration can be very useful in eliminating the need to query each data source separately when collecting information (Clifton et al., 2004). In addition, the results of integration can be shared among different data consumers.

[1]XML: Extensible Markup Language http://www.w3.org/XML/
[2]RDF: Resource Description Framework http://www.w3.org/RDF/
[3]OWL: Web Ontology Language http://www.w3.org/2001/sw/wiki/OWL

Data is usually provided by organisations or entities through *data sources* or *datasets*. Data sources may have several dimensions including differences in: data organisation models, representation, scope, abstraction, meaning, and temporal validity (Wiederhold, 1993; Ullman et al., 2001). Data sources participating in any data integration process can be homogeneous or heterogeneous. However, to generate information it is essential to combine heterogeneous data sources (Wiederhold, 1993). Therefore, data integration approaches usually consider the heterogeneity of the data in order to provide the users with suitable responses to their queries.

The integrated data is normally the result of queries requested by data consumers. Data can be consumed in two main formats, human readable, for example web browser tables, Microsoft Word documents and charts, or machine and parsable formats such as: CSV[4], XML, JSON[5], and Microsoft Excel. Systems built on the basis of consumers querying several heterogeneous databases are the so-called Data Integration Systems (DIS) (Dicelie et al., 2001; Gusmini and Leida, 2011; Guo and Fang, 2011).

2.1.2 Integration Approaches

To achieve a true integration between different data sources, a well-defined data integration approach is needed. The literature suggests that data integration approaches are either through the method of integration or the data location, see Figure 2.2.

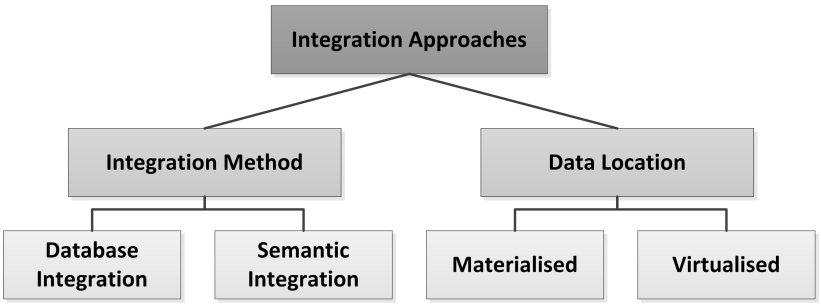

Figure 2.2: Integration Approach Categories

Based on the method of integration: the two main approaches are database integration and semantic integration (Noy, 2004). Database integration is based on resolving a user's query against a global schema that represents a set of relational databases (Lenzerini, 2002; Noy, 2004). Ullman et al. (2001) proposed three common integration approaches using databases: federated databases, a centralised warehouse, and the use of a mediator. Federated databases are slightly different from the other types as they involve a one-to-one simple architecture in which databases pair to talk to each other and execute queries. The centralised data warehouse extracts data from databases and combines them using a global schema, making this global schema appear as a single database. The mediator

[4]CSV: Comma Separated Values https://www.w3.org/2013/05/lcsv-charter.html
[5]JSON: JavaScript Object Notation http://www.w3schools.com/js/js_json_intro.asp

approach provides a global view that integrates several views of the databases. It is similar to the data warehouse except that it does not store the data itself.

Semantic integration is achieved by using semantic matches to answer users' queries. This uses a mediator approach. One example is by using a local schema for each data source and a global mediated schema for the integration. For any query directed to the mediated schema, the system translates the query into a set of semantic matches that run between the mediated schema and the local schemas. Each data source wrapper[6] executes its part of the query and the results are then combined and presented to the user (Doan and Halevy, 2005). Semantic integration is considered one of the serious challenges in data integration (Noy, 2004; Doan and Halevy, 2005). There are many reasons for this difficulty: one is due to the heterogeneity of data, while others involve the schema matching process that finds semantic matches between the database schemas (Doan and Halevy, 2005).

In addition, the integration methods can be expanded to include the integration of NoSQL databases. NoSQL stands for *Not Only SQL* or *Not Relational*, but is not agreed upon according to Cattell (2010). There are three types of NoSQL databases: key-value stores, document stores, and wide column stores (Cattell, 2010; Leavitt, 2010). These databases are non-relational and hence include hierarchical, graph, and object-oriented databases, and are known since the 1960s (Leavitt, 2010). The NoSQL databases are widely used in big data applications and social media where scalability and complexity are expected and the queries to them can be conducted via web APIs. These databases can be combined together or with other types of database as demonstrated by Lawrence (2014) that presents a generic architecture to allow the integration of SQL with NoSQL databases.

Based on the data location: these approaches can be divided into materialised and virtualised. The materialised approach is the traditional data integration that relies on the Extract, Transform, Load (ETL) process, which requires caching the data into a mediator (i.e. a data warehouse) for transforming and using it (Gupta and Mumick, 1995; Loshin, 2010). Although it is considered efficient in answering queries, this approach is costly in maintaining an up-to-date copy of the data source (Calvanese et al., 1998).

The virtualised approach is new and uses data virtualisation, where data remains within its sources and is materialised when needed (Hull and Zhou, 1996; Loshin, 2010). Despite the efficiency of this approach in saving time and space on the mediator side (Zhou et al., 1995), it is costly to access data sources (Calvanese et al., 1998) to answer each query.

Choosing the appropriate approach depends on how often the data changes. The virtual approach is better if the data changes frequently, if not, then the materialised approach

[6] A wrapper is an extractor/adaptor that connects to a data source to translate queries and results between the global schema and the local schema of the source (Ullman et al., 2001).

is suitable. However, these approaches can be combined to capitalise on the benefits of each (Hull and Zhou, 1996).

2.1.3 DIS Architecture

DIS architectures usually contain several components. These differ in their naming and functionality, depending on the requirements of the application. For example, in an ontology approach to data integration, Gusmini and Leida (2011) propose a DIS that includes a number of data sources, a global ontology comprising several elements, concepts and attributes, a mapping system between data sources and the global ontology, and a user interface. Another example is the system proposed by Dicelie et al. (2001) that aims to integrate legacy data sources with different formats and access methods. It contains a back-end interface for proper input and output data conversions (i.e. converting to XML or HTML[7]), and a middle tier for conversion rules management.

Nachouki and Quafafou (2011) proposed a web-based DIS of three layers: data sources, a mediator, and agents (i.e. users or applications). The mediator acts as an interface between data sources and agents and contains the mediation schema. One drawback of such architectures is that they depend heavily on an administrator to control the global and mediated schema, which is a very difficult task in the web environment. Langegger et al. (2008) also used a mediator-wrapper architecture in a virtual semantic integration. The wrappers are used to map data sources to the common schema processed by the mediator.

Guo and Fang (2011) proposed an ontology-based DIS that aims to resolve queries dynamically and addresses the changeable needs of a system. The DIS contains a heterogeneous data layer that contains data sources utilising web services, a middleware layer, and a user view that shows the results depending on users' queries, see Figure 2.3.

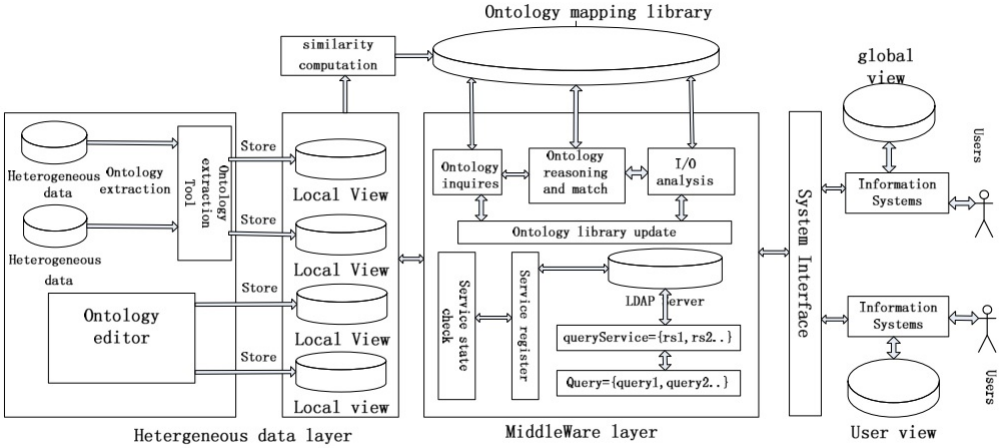

Figure 2.3: Architecture of an Ontology-based DIS by Guo and Fang (2011)

[7]HTML: HyperText Markup Language https://www.w3.org/MarkUp/

In multi-layer DIS, the location where the integration is achieved has different names depending on the integration approach and the integration context. Some studies refer to the location as the mediation engine (Bhowmick et al., 2006) or the mediator (Cruz et al., 2008). The location is basically used for either materialising the data to create a global schema, or for virtualising the data to create a global view.

2.1.4 Data Integration Applications

Using multiple heterogeneous sources to produce data has been used in many applications. In the 1990's, data integration moved from research labs to industry in the form of Enterprise Information Integration (EII) that focused on integrating data sources without materialising (i.e. caching) the data in a central warehouse. Several factors allowed this development, such as the maturity of some of the integration technologies, the changes in the need for data management in organisations, and the emergence of XML (Halevy et al., 2005, 2006).

Data is integrated to serve different purposes. It has been used in visualisation (Keim et al., 2006), analysis, and decision-making in many areas, such as business intelligence (Dayal et al., 2009). It has also been used generally in scientific research (Langegger et al., 2008), biology research (Ray et al., 2009), healthcare (Bhowmick et al., 2006), web of data (Bizer et al., 2009), and national security (Harris et al., 2007).

On the web, integration has been introduced as the composition of web services using Web APIs in so-called data mashups. Data Mashups are a specific breed of web service that are based on combining data from different resources (Fung et al., 2012). In mashups, data sources are accessible through web services using REST[8] or SOAP[9], HTTP[10] protocols, and XML RPC[11](Lorenzo et al., 2009).

In the context of cloud computing, Carey et al. (2012) discussed how data services are introduced to provide a more accessible and richer model to the data consumers, differing from traditional web services. Data integration is provided as a cloud service, namely integration Platform as a Service (iPaaS) (Palanimalai and Paramasivam, 2015). Data services can integrate heterogeneous data sources coming from other data services, cloud data, web services, and relational databases, to answer users' queries. Several domains benefitted from data integration cloud services, for example integrating and analysing healthcare information coming from diverse geographical locations using the cloud (Bahga and Madisetti, 2015). Moreover, traditional data integration has shifted into Big Data Integration (BDI) where the volume, velocity, variety, and veracity of the data are changed (Dong and Srivastava, 2015).

[8]REST: Representational State Transferhttps://www.w3.org/2001/sw/wiki/REST
[9]SOAP: Simple Object Access Protocolhttps://www.w3.org/TR/soap/
[10]HTTP: Hypertext Transfer Protocol https://www.w3.org/Protocols/
[11]XML RPC: XML Remote Procedure Calling protocol http://xmlrpc.scripting.com/spec.html

2.2 Data Leakage in DIS

Integrating personal and sensitive data sources in DIS incurs the threat of data exposure, as introduced in Section 1.1. The threat of exposing the data falls under a generic security threat named data leakage. The following sections give an overview of security, privacy and trust, then a definition of data leakage and its causes in the context of data integration.

2.2.1 Overview of Security, Privacy, and Trust

Information Security, as defined by the ITSEC[12] and ISO[13], is a combination of confidentiality, integrity, and availability (European Communities-Commission, 1991; ISO, 2014). One can argue that this definition is incomplete, as many aspects of security, such as authenticity, are not included (Gollmann, 2006). This is because security is system dependent; a security property of one system may not be a security property in another (Wang and Wulf, 1997), depending on the different requirements and business needs. However, security properties include authenticity, authorisation, confidentiality, integrity, availability, and non-repudiation (Walton et al., 2009). Security properties are inter-related, which means that one aspect influences another; for example, authentication relies on non-repudiation (Walton et al., 2009).

The purpose of information security is to protect assets and mitigate risks. Several terms need to be defined to help understand the nature of information security. A *vulnerability* is a "*weakness of an asset or control that can be exploited by one or more threats*" (ISO, 2014). A *threat* is a possible action or event that may harm a system or organisation and be detrimental to its security (European Communities-Commission, 1991; ISO, 2014). The actual attempt to harm a target by destroying, exposing, changing, disabling, stealing, or gaining unauthorised access or use of an asset, is called an *attack* (ISO, 2014). Finally, a *risk* is the "*effect of uncertainty on objectives*" (ISO, 2014).

As one of the properties of security, **confidentiality** aims to prevent the disclosure of information to unauthorised individuals, entities, or processes (European Communities-Commission, 1991; ISO, 2014). One of the ways to enforce confidentiality is by the use of cryptography (Watson, 2007) to encrypt confidential data. In addition, it can be enforced by implementing access controls that are considered a way to restrict access to assets to authorised parties only depending on business and security requirements (ISO, 2014).

Privacy is concerned with the protection of personal information (Gollmann, 2006) and determining when, how and what type of information can be exposed to others

[12]ITSEC: Information Technology Security Evaluation Criteria
[13]ISO: International Organisation for Standardisation http://www.iso.org/iso/home.html

(Jawad et al., 2013). It is also defined by Westin (1970) as "*the right of an individual to decide what information about himself/herself should be communicated to others and under what circumstances*". Privacy has been addressed in legislation, such as the UK DPA[14] and is usually covered in an organisation's security policies.

Privacy should be investigated separately from security. The reason is that, although a user may have an appropriate authorisation to access the data, which satisfies security, this does not prevent a secondary analysis of the authorised data, which potentially violates privacy (Bhowmick et al., 2006). In addition, in the context of integration, the displayed integrated results may reveal an individual's identity, which also violates privacy.

Data privacy has several dimensions. Jawad et al. (2013) claim that it covers "*collection limitation, purpose specification, use limitation, data quality, security safeguards, openness, individual participation, and accountability*". Constante et al. (2013) argued that privacy has: 1) purpose: reasons for data collections and usage, 2) visibility: entities allowed to disclose the data, 3) retention period: duration for maintaining the data, and 4) data sensitivity: the subject's perception of harm that can be caused by data misuse.

Trust is a relationship between several entities that is based on assumptions, which, when satisfied by an entity, mean it is considered trustworthy (Viega et al., 2001). From a security perspective, trust is the belief that an entity, i.e. a system, process, or person, will act in a certain way when the security policy is enforced (Ross et al., 2014).

There are many ways to determine the level of trust in an entity. Artz and Gil (2007) suggested policy-based trust, reputation-based trust, or adopting a general trust model. In terms of trust models, trust can be expressed as a range of levels or degrees to determine the trustworthiness of an entity (Ross et al., 2014).

2.2.2 Data Leakage Definition

Data leakage is the disclosure of confidential information to unauthorised entities intentionally or unintentionally (CWE, 2013). Any system can display an acceptable amount of information, such as revealing the last 4 digits of a credit card; therefore, not all data leakage is serious (Chothia et al., 2013). However, data leakage is a major threat to any information system that handles confidential data. Leakage can occur at: system level (Chatzikokolakis et al., 2010), network level, such as in web traffic (Borders and Prakash, 2009), code level (Zanioli et al., 2012), document level (Gessiou et al., 2011), and data level (Kaufman et al., 2011).

This study focuses on leakage that occurs to personal and sensitive data. Since the main purpose of DIS is to provide data to the consumers, as discussed in Section 2.1,

[14]DPA: Data Protection Act http://www.legislation.gov.uk/ukpga/1998/29/contents

the threats investigated here are those relevant to the data itself rather than those related to the platforms, operating systems, networking infrastructure, and software code.

Data leakage threats in DIS are a result of data exposure by unauthorised access, by escalation of privileges, by lack of understanding of the value of the data, or by employing poor security techniques. Therefore, when it occurs, data leakage breaches the confidentiality of the data and violates privacy. It can originate from authorised and trusted entities as well as untrusted ones. Hence, confidentiality, privacy, and trust are the properties chosen as the focus for this study.

2.2.3 Causes of Data Leakage in DIS

The following sections explain some of the possible causes of the violation of confidentiality, privacy, and trust that lead to data leakage.

2.2.3.1 The Lack of Confidentiality and Privacy Techniques

Violation to *confidentiality* that leads to data leakage can occur by failing to implement access control models and encryption techniques. In addition, the inability to handle sensitive data within the integration process in a correct manner leads to such violations.

Flaws in *access control models* employed lead to unauthorised access. This can occur by: 1) the use of an inappropriate access control model (Braghin et al., 2003), 2) the weakness of the selected model (Watson, 2007; Tipton and Nozaki, 2007), and 3) the misconfiguration of the model (Pistoia et al., 2007). In some cases, multi-level access controls utilised by Mandatory Access Control (MAC) cause data leakage. As an example, assume only two levels of security. The operating rules on these might prohibit the lower level reading or accessing the higher level, while the higher level is prohibited from writing or leaking data to the lower level. However, if they have a shared resource that may be compromised by a Trojan horse, this may cause writing or leaking data to a lower security level (Braghin et al., 2003).

In an access control model, misconfigured policies may cause data leakage by overlapping roles and accidently granting permission to access sensitive data. Therefore, such models need to be continuously evaluated. Pistoia et al. (2007) discussed how Role-Based Access Control (RBAC) policies may be insufficient, redundant, and even subversive, and proposed an approach to evaluate them.

In addition, the lack of *data encryption* techniques can lead to information disclosure. Herbert and Thieme (2012) proposed the use of encryption to protect sensitive data from leaking to the platform during the integration process.

When *handling sensitive data in the integration process*, underestimating the implications of inapplicable confidentiality leads to data leakage. This can be caused by the inconsistencies that occur between regulatory laws and uncertainties in data ownership, access rights, and disclosure, in cases where data is passed between different countries/states (Meingast et al., 2006). Another example is the confidentiality of data not being preserved when data is being merged (Batty et al., 2010). In addition, users' ignorance of legal issues in data management allows unauthorised access to occur (Batty et al., 2010).

Bhowmick et al. (2006) claimed that privacy in data integration is even more complicated than security. The threat of data disclosure to unauthorised parties, according to Caceres and Teshigawara (2010), is a threat that is widely ignored and is caused by information eavesdropping or privacy invasion. Several violations to privacy elements lead to data leakage as summarised below.

Since one of the purposes of privacy is to protect personal information, the disclosure of individuals' identities is a form of data leakage, depending on the context. Identities can be exposed by disclosing attributes that directly identify a person, namely Personal Identifiable Information (PII), such as name and social security number (Guarda and Zannone, 2009). Another way to expose an identity is by disclosing attributes that can be used to re-identify individuals uniquely (Sweeney, 2002), called Quasi IDentifiers (QID), such as gender, birth date, and postcode (Mohammed and Fung, 2010).

Li et al. (2013) discussed two types of privacy attack in DIS, query attribute correlation and inference attack. *Query Attribute Correlation* attacks aim to intercept queries and analyse predicates, query location, query content, data location, and expressions, to correlate attributes and infer sensitive information.

Inference Attacks are very common in DIS. One type is achieved by record linkage, where links are made between the data and non-confidential information or statistical aggregates to infer private data (Clifton et al., 2004; Zhang et al., 2011). Another type is achieved by attribute linkage, through linking QID attributes together (Fung et al., 2012) or with external data (Whang and Garcia-Molina, 2012).

A further type of inference attack is the use of consecutive queries, where consumers issue multiple queries to infer information (Bhowmick et al., 2006; Clifton et al., 2004). Consumers are able to use the results of the queries to conduct inference attacks, such as inferring confidential data from non-confidential data or from statistical aggregates, as cited by Zhang et al. (2011). Boyens et al. (2004) defined another form of inference attack, interval disclosure, which is the ability to compute missing sensitive values using the aggregate information already published by the mediator.

There are cases when there is a need to integrate data from public data sources with an organisation's private data sources. This integration leads to different challenges,

such as the lack of privacy measurement, i.e. measuring the loss of privacy when data is exposed (Pon and Critchlow, 2005). Yau and Yin (2008) proposed a repository for data integration across data sharing services by collecting data based on users' requirements. If the repository is compromised, only the result of the integration is revealed and the original data will remain intact. Mohammed et al. (2011) proposed two algorithms from game theory to overcome the challenges of revealing data coming from different data providers.

Data leakage can arise from insider attacks. This occurs when data providers use their data, which is a subset of the whole data, to infer other data records contributed by other data providers (Goryczka et al., 2013). Data exposure is also caused by using the data contrary its intended purposes. Clifton et al. (2004) suggested matching the purpose statement of the data consumers with the privacy policy of the data sources, which requires the annotating of data sources with privacy meta-data. Guarda and Zannone (2009) also emphasised the need to check and match the purpose of data collection and use.

2.2.3.2 The Use of External Entities

The use of cloud services or third-party entities to assist in the integration process, exacerbates security threats. These entities may not protect the data according to its security policies or may allow transitive trust to other entities.

Clouds are critical to security and privacy and require further effort to increase their reliability (Saeed et al., 2014). Clouds need to comply with regulatory laws for data protection (Takabi et al., 2010; Youssef and Alageel, 2012). There are several risks to security and privacy in the cloud (Saeed et al., 2014; Carey et al., 2012; Hashizume et al., 2013). The risks arise from using multi-tenancy public clouds that share physical infrastructure with untrustworthy users, which leads to attacks such as cross-virtual machine attacks (Ristenpart et al., 2009). To prevent unauthorised access, clouds need to deal with the heterogeneity of security components and multi-tenancy (Takabi et al., 2010).

Using cloud services to process and integrate data, in addition to providing services to query and analyse the data (Carey et al., 2012), is prone to risk. Trust in cloud providers and servers needs to be present, otherwise clients may find their data censored or used to make profits (Jawad et al., 2013). The fact that the clouds are not under an organisation's physical control exacerbates the problem of managing data security (Reeve, 2013), especially when there are no standard rules and regulations to deploying the cloud (Saeed et al., 2014).

Third party entities are used in data integration applications for different purposes. Organisations may want to outsource the data to a third party to analyse it and compile

aggregation statistics (Xiong et al., 2007) or to handle data backup (Hashizume et al., 2013). An organisation may require an entity to handle access control to personal integrated data, such as the approach described by van den Braak et al. (2012) that uses a trusted third party to handle access controls to government data in the public sector.

Data leakage is possible through third party entities due to the lack of clarification on these entities' rights to the data (Meingast et al., 2006). In addition, the lack of security policies defining how transitive trust is achieved from one party to other different parties (Fung et al., 2012) increases the risks of data leakage. Therefore, data transfer to clouds and third parties needs to be based on trust (Saeed et al., 2014).

2.2.3.3 The Distribution of DIS Components

Distributed DIS components (i.e. data sources, middle layers, data consumers) can aggravate security threats as the mere nature of a distributed system encourages that. In distributed DIS, data travels from its sources through several layers to reach the consumers. Hence, the data and its policies may suffer exposure at any layer if security measures are not in place. Any distributed system is prone to well-known attacks, such as eavesdropping, denial of service, replay attacks, message tampering, and masquerading (i.e. guessing user's identity) (Prakash and Darbari, 2012).

Manan et al. (2011) discussed security concerns in a general distributed environment as discussed in Section 1.1. However, Huang et al. (2008) argued that security problems can also arise in a DIS that uses centralised data integration using standard JDBC/ODBC[15] to access the data. They reasoned that firewalls are used to protect data sources and the integration location, and therefore integrating data from outside requires opening new ports in the firewall, leading to security threats.

This study covers DIS with a middle layer, where data leakage can occur in any layer because each layer can be in a different physical or logical location. Hence, data leakage threats can be aggravated by the very existence of these layers.

2.2.3.4 Assuming Trust in DIS Components

Violating trust leads to data leakage in DIS. This occurs when an entity behaves contrary the agreed security policy (Ross et al., 2014). Generally, trust is an issue in distributed systems (Prakash and Darbari, 2012). The DIS entities that are susceptible to violating trust are the mediator (or the integration location) that can even be external to the system, as discussed in Section 2.2.3.2, and the data consumers.

[15] JDBC/ODBC: Java Data Base Connectivity/Open Data Base Connectivity

Violations by data consumers are possible when the system assumes they are trustworthy. The lack of consideration of the risks associated with consumers in handling sensitive data can result in secondary use of data (Paci et al., 2013). Once the data has been disclosed to consumers, they could misuse the data by sharing it with unauthorised parties, or use it for purposes other than the use intended by the data provider.

2.3 Mitigating Data Leakage in DIS

Many studies have discussed the mitigation of security and privacy challenges in DIS. Due to this diversity, the topics have been organised according to themes (Cronin et al., 2008) in the following sections. This approach enables the research gap to be clearly identified and establishes the need for this study.

2.3.1 Securing Data Sources

Data-centric security is mainly concerned with the data itself and its ability to be protected, regardless of the application or system it resides in (Takabi et al., 2010). The IBM Data Centric Security Model (DCSM) suggests the use of security meta-data that provides more information about the level of data sensitivity. The meta-data can be used to give privileges to users according to their role in the system, with proper access controls (Hennessy et al., 2009). However, the model does not consider the issue of combining meta-data, in case of data integration, and the implications of the combination on users' roles.

Security meta-data provides complete information about data provenance; however, it needs to be balanced against data privacy (Takabi et al., 2010). It is not clear how data privacy is maintained, nor how the data can be protected in outsourced computations (Takabi et al., 2010). The use of security meta-data to enforce authorisation in data warehouses has been suggested by Katic et al. (1998). Several security meta-data fields can be included with the data sources to be integrated in DIS to ensure the security and privacy requirements of the sources are maintained. Table 2.1 shows a synthesis of these fields from several studies.

Data sources containing sensitive data need to explicitly indicate its sensitivity and the ways to handle it. Privacy dimensions, as discussed in Section 2.2.1, can be part of the privacy requirements associated with data sources. Arenas et al. (2010) discussed a data sharing agreement contract that "*regulates who can access data, when and where, and what they can do with it.*" Clifton et al. (2004) suggested annotating data sources with privacy meta-data to express the privacy policies that need to be applied during the integration.

Table 2.1: Synthesis of Data-centric Security Meta-data

Data-centric Security Meta-data	**Author(s)**
What data is used in the system?	Hart (2001)
Who owns the data? i.e. who deletes, modifies and enforces regulations.	Meingast et al. (2006); Hennessy et al. (2009); Takabi et al. (2010)
What is the data retention period?	Hennessy et al. (2009)
How is the data classified in terms of sensitivity level? (Such as intellectual property, trade restriction, or privacy properties)	(Hart, 2001; Hennessy et al., 2009)
Are data protection mechanisms appropriate for the required level of sensitivity?	Hart (2001)
Who will access the data? (Internal and external entities)	Hart (2001)
What access control is used?	Hart (2001); Hennessy et al. (2009); Takabi et al. (2010)
What are the encryption requirements for the data in storage or in transit, according to the organisation's policy?	Hart (2001); Takabi et al. (2010)

2.3.2 Privacy Preserving Data Integration

Data integration approaches, as discussed in Section 2.1.2, have different characteristics that may influence the ways that can be used to secure them. They have specific approaches to the integration of materialised data. These aim to create secure data warehouses, which differ from the ways of securing virtualised approaches that do not store data.

Some data integration approaches are designated as privacy-conscious, where specialised techniques are used, such as anonymisation (Mohammed et al., 2011), generalisation, suppression (Sweeney, 2002), and data perturbation (Chung, 1993). Anonymisation is removing identification information from the data (Mccallister et al., 2010), while perturbation is adding noise to the data to achieve privacy (Mohammed and Fung, 2010). In these approaches, removing and replacing identity attributes by other attributes can be used (Meingast et al., 2006).

Many studies have been published on privacy-preserving data integration for either materialised or virtualised approaches. These studies provide technical approaches, namely privacy-preserving techniques in data mining and integration to achieve data privacy in DIS and minimise the threat of privacy violation (Clifton et al., 2004; Pasierb et al., 2011; Pon and Critchlow, 2005; Boyens et al., 2004; Xiong et al., 2007).

Combining data may violate privacy and so inference attacks and attribute correlations are among several privacy attacks discussed in Section 2.2.3.1. To enforce privacy, privacy-preservation needs to operate at a deeper level in mining and integration methods. Privacy-preservation can be carried out through materialised integration approaches, such as the use of data warehouses. Boyens et al. (2004) proposed an approach to mine data in warehouses, in mediator-based data integration, while applying privacy-preserving techniques through anonymisation that aim to prevent interval disclosure attacks. Integration approaches have advanced so that they can integrate private databases to achieve information sharing while preserving privacy (Wang et al., 2005).

Data integration's focus on privacy has arisen primarily from healthcare domains, where the privacy of patients is important and protected by law in several countries, such as HIPPA[16] in the United States. One example is integrating healthcare data from several organisations and maintaining patient privacy through anonymisation, as proposed by Mohammed and Fung (2010).

2.3.3 Security and Privacy across the DIS

Another approach of applying security to DIS is achieved by considering the security and privacy of all the components of DIS or several of them together at the same time. Few publications have considered all components, but are among the most relevant to this research.

van den Braak et al. (2012) proposed a framework that supports data sharing among different public organisations. It preserves privacy through sharing data based on the need-to-know principle, where data is provided only when required for a specific process. The authors propose the notion of a Trusted Third Party (TTP). The TTP is responsible for integrating and sharing data between different government organisations. The proposed framework contains two parts: the first is data integration techniques to achieve privacy, while the second provides guidelines on data sharing that ensures security and trust. The guidelines mainly focus on the integration location and the data sources, rather than the system as a whole. However, the integration was across government organisations, and thus the sources were under the same types of security policy. Therefore, the risk of violating the integrated security policies was absent.

The approach proposed by Clifton et al. (2004) is a privacy framework for data integration. The purpose of the framework is to provide an insight into the privacy challenges in data integration. It provides several research directions, one of which emphasises the need for a privacy framework that considers users' views on exposing and hiding sensitive attributes. It proposes privacy policies implementing users' views and a purpose statement specifying which data is allowed to be accessed and integrated. The

[16]HIPPA: Health Insurance Portability and Accountability Act http://www.hhs.gov/hipaa/

framework aims to preserve privacy in data integration in view of the data sources, integration approach, data consumers, and the security policies. The authors addressed data leakage mainly by discussing the difficulty of preventing multiple query attacks. The heterogeneity in the security and privacy policies are only briefly addressed and there is no specific focus on them and their relation to the integration approach.

Bhowmick et al. (2006) is similar to that of Clifton et al. (2004), but it proposed a more detailed architecture and framework for privacy-preserving data integration and sharing deployable DIS. It includes security and policy considerations by adding a security policy component to the system. It also provides several suggestions on preserving privacy in different DIS components. The proposed architecture covers most of the DIS components. However, the level of detail provided for integrating the security policies of the data sources and the integration location is insufficient.

Jurczyk and Xiong (2008) focused on privacy-preserving data integration. It proposed several protocols for data anonymisation in addition to a general architecture for data integration. Hu and Yang (2011) proposed a semantic privacy-preserving model employed on multiple servers for data sharing and integration that provides authorised query answering.

2.3.4 Coverage of Confidentiality, Privacy, and Trust

The focus of studies on the properties of confidentiality, privacy, and trust varies. Some tackle each property separately; others combine two of them. Only a limited number of studies have focused on the three properties combination in the DIS context.

On *confidentiality* in DIS, studies by Haddad et al. (2012) and Begum et al. (2010), have focused on security policy integration and conflict reconciliation and their uses to answer users' queries. Other studies have proposed extensions and improvement to RBAC to adapt to the integration context, such as Lamb et al. (2006).

Privacy in DIS has the lion's share of research. Privacy-preserving techniques are well-established in the literature, spanning a range of different topics from privacy in peer-to-peer DIS to anonymisation techniques. Bhowmick et al. (2006) proposed a privacy-preserving DIS framework that emphasises the need to consider the balance between privacy and data sharing. This perspective has been addressed by many later studies: Pasierb et al. (2011) presented different approaches to privacy-preserving data integration in e-healthcare systems, while Barhamgi et al. (2011) applied the same concept to web services and data mashups.

On *trust* in DIS, many distributed trust models can be used to determine the level of trustworthiness of systems' entities. Treglia and Park (2009) suggested a trust framework for intelligence information sharing between agencies. Other approaches focus on

computational trust using either policy-based trust or reputation-based trust (Artz and Gil, 2007) that can also be applied to DIS. Prakash and Darbari (2012) discussed several security approaches to enforce trust, such as trust models and risk management as a method to evaluate trust.

Other studies acknowledge the need for a combination of properties; for example Hung (2005) combined security and privacy by extending the RBAC model with privacy-based extensions.

2.4 Software Security

Software security is a recent field and McGraw (2004) argued that there is no clear best practice for it yet. The software engineering community has realised the need to consider software security after noticing the significant impact unsecure software had in critical domains, such as national security. Some companies have lost their reputations and suffered significant financial crises due to security breaches. Therefore, having secure software that continues to function correctly, even under malicious attack, has become a necessity (Chandra and Khan, 2008).

The following sections explain ways to achieve secure software. Starting from the concept of security and privacy by design, the importance of threat analysis in securing software is explained. Security guidelines, policies, and access controls are then discussed. Finally, the need for formal analysis of security policies is discussed.

2.4.1 Security and Privacy by Design

One of the common approaches to achieve security by design is by including security in the Software Development Lifecycle (SDLC). The SDLC consists of stages of software development until it is ready for release. McGraw (2013) emphasised the importance of designing security within the development process, which he advocated earlier in (McGraw, 2004). He considers cyber security as an important context and points out that, as practitioners become aware of its importance, they are now progressively using and advancing best practice to address the problem of software security.

It is essential to define security goals and objectives, depending on the context of the information system. Haley et al. (2008) suggested that security goals can be expressed in operational terms by clearly-stated security requirements that can be given to a designer and an architect to apply in the designed systems. They also believe that security recommendations should be considered as security requirements. Therefore, these requirements should be derived from already defined security goals.

Security requirements can be added to every step of the development lifecycle (Devanbu and Stubblebine, 2000; McGraw, 2004). The Microsoft Security Development Lifecycle (SDL) and Adobe Secure Product Life Cycle (SPLC) both provide security activities that are applied to the development approach (Chess and Arkin, 2011). Software security needs to be maintained over the full lifecycle, as security issues can arise from any part of the system (McGraw, 2004).

Many examples exist of how security can be included in the development lifecycle: Mouratidis et al. (2005) suggested the "*security-oriented paradigm*" that focuses on defining security in the early requirement and design phases are "*propagated*" in later development stages. Apvrille and Pourzandi (2005) took a more comprehensive perspective: they suggest security techniques for every step of the development (except maintenance). This led to Chandra and Khan (2008) who proposed an object-oriented software security estimation lifecycle. This lifecycle was implemented in a framework later in (Chandra et al., 2009) to include the early stages of security estimation activities. It is challenging to create absolute and clear security requirements specifications (Mir and Quadri, 2012), yet it is still worthwhile gathering these to be delivered to the next development phase.

Similarly, privacy can be included in system development by design. Guarda and Zannone (2009) discussed the dimensions of privacy relevant to data protection regulations to help build privacy-aware systems. Schaar (2010) suggested considering privacy in all IT systems that handle personal data, and that systems should avoid or minimise the amount of personal data processed to enforce privacy by design. Gurses et al. (2011) discussed two case studies where the principles of privacy, including personal data minimisation, are employed to ensure privacy by design.

In general, there is a growing body of knowledge on methods of integrating security and privacy in software development. Some provide generic security activities in every phase of the development, a survey of which can be found in (Khan and Zulkernine, 2009), that can be complemented by the use of security standards and guidelines to secure software.

Considering security in software design is equally important to managing and monitoring the implementation of security techniques. One of the deadly sins of security management, according to von Solms and von Solms (2004), is "*not realising that information security compliance enforcement and monitoring is absolutely essential.*" Security management covers many tasks but those relevant to this study are concerned with technical management, such as auditing the compliance with policies and legal requirements (Youssef and Alageel, 2012), monitoring data deletion and modification, and having appropriate backup to prevent data loss (Youssef and Alageel, 2012).

2.4.2 Threat Analysis and Modelling

How should security be integrated in the SDLC and risks mitigated? The attacks, threat models, and attack patterns must first be understood and then included in the development (Whittaker and Howard, 2004; McGraw, 2004). This is not a simple task, given the complexity of applications and the level of hacking expertise. It is not possible to write a 100% secure, fault-free program. Security is a complicated topic that can be divided into smaller parts, and training developers on how to tackle those parts can be achieved with a methodical approach (Apvrille and Pourzandi, 2005).

One approach is to conduct threat analysis and modelling prior to software development. Threat modelling is the systematic approach that identifies threats and determines the appropriate countermeasures (Meier et al., 2003). Both threat analysis and identification can be conducted using tools. Two examples of popular techniques that help system designers identify security and privacy threats are Microsoft's STRIDE (Torr, 2005) and LINDDUN (Deng et al., 2011).

STRIDE provides a taxonomy of the type of threats. It is the acronym of: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. Each of these threats negate a security property: confidentiality, integrity, availability, authentication, authorisation, and non-repudiation. STRIDE guides system designers through a systematic process to the identification of security threats. First, a model of the system is created and its components are mapped to the six threat categories. Then, a catalogue of threat tree patterns is used to identify specific instances of threat categories, and the level of risk for each threat is determined. Finally, the risk of the threat is reduced or eliminated by introducing proper countermeasures and defences. Once threats are identified, the security requirements of the system can be based on them (Myagmar et al., 2005).

LINDDUN follows a similar process to help systems designers identify privacy rather than security threats. LINDDUN provides a taxonomy of privacy threats that violate specific privacy properties. It includes an extensive catalogue of specific threats. For each category of threat, a list is provided of privacy-enhancing technologies that mitigate those privacy threats. LINDDUN focuses on these to elicit system requirements: Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, and Non-compliance.

Since many vulnerabilities are caused by flawed software design (McGraw, 2004), using the previous approaches or similar tools are very useful in considering threats before the actual development of the system, which ensures security and privacy by design.

2.4.3 Security Guidelines

To approach the general security of software, one or more of the following can be used: policies, standards, guidelines, and procedures. Wood (2005) differentiated between security policies and security standards. Security policies define the high-level and mandatory general management instructions that specify an arranged course of action for handling a problem. Security standards differ from policies as they provide mandatory mid-level technical requirements, including implementation steps, design concepts and other specifics. Cannon (2008) linked the previous concepts with guidelines and procedures. In cases where security standards are absent, security guidelines provide best practice and are strongly recommended. However, security procedures are more detailed step-by-step actions to support standards.

The security guidelines are often a clear set of elements that assist software engineers in developing secure software. At the beginning of development, security requirements are identified, which is helpful in implementing the proper security controls and mechanisms (Futcher and von Solms, 2008). Although using security guidelines, and therefore security features, is very useful in building secure software, failure to adopt a security strategy and implementing it correctly can be very dangerous (such as not knowing how to integrate SSL[17] properly) (Apvrille and Pourzandi, 2005; Futcher and von Solms, 2008). Many security issues arise from the fact that developers lack computer security knowledge (Juerjens, 2002).

Security activities and guidelines can be linked to security standards, such as the SDLC model developed by Horie et al. (2009). The available international security standards used currently are provided by key organisations, such as IEEE[18], NIST[19], ISO, ANSI[20], DOD[21], BSI[22], and OMG[23]. The CORBA[24] architecture is also a widely used standard (Futcher and von Solms, 2008). There are other bodies relevant to information security, such as the IEC[25] and the ITU[26] that aim to produce different security standards. These standards are originally fixed guidelines "*analysed by someone authorised, experienced, and qualified in the specified area*" (Futcher and von Solms, 2008).

Security guidelines are constructed in different ways. Table 2.2 summarises five examples of guidelines that mainly rely on the existing literature, or on security standards and concepts, as sources of information.

[17] SSL: Secure Socket Layer
[18] IEEE: The Institute of Electrical and Electronics Engineers https://www.ieee.org/index.html
[19] NIST: National Institute of Standards and Technology https://www.nist.gov/
[20] ANSI: The American National Standards Institute https://ansi.org/
[21] DOD: The US Department of Defencehttp://www.defense.gov/
[22] BSI: The British Standards Institution http://www.bsigroup.com/
[23] OMG: Object Management Group http://www.omg.org/
[24] CORBA: The Common Object Request Broker Architecture http://www.corba.org/
[25] IEC: International Electro-technical Commission http://www.iec.ch/about/
[26] ITU: International Telecommunications Union https://www.itu.int/en/Pages/default.aspx

Table 2.2: Some Approaches to the Creation of Security Guidelines

Authors	**Type of Guidelines**	**Construction Method**
Jaferian et al. (2008)	Design guidelines for IT security management tools	Surveys the literature for guidelines and features, classifies guidelines, and links guidelines to challenges in the field.
Kasemsan and Hungam (2011)	Internet banking security guideline model for banking in Thailand	Reviews the literature for guidelines, surveys a sample of 400 people to evaluate the guidelines and constructs the model.
Nurse et al. (2011)	Guidelines for usable cyber security	Reviews the literature, groups and analyses the guidelines.
Futcher and von Solms (2008)	Guidelines for secure software development	Analyses the key security and software development standards to create a number of overlapping activities presented as guidelines.
Heikkinen et al. (2009)	Security and user guidelines for the design of the future networked systems	Combines secure technical design concepts based on classical general principles and user experience with security to create guidelines.

Harris et al. (2007) discussed standard-based approaches to secure data across organisations in data integration applications that handle and share critical data, such as emergency response and healthcare awareness. Their work emphasised the need to enforce security policies and create standards or guidelines to govern these types of applications.

2.4.4 Security Policies and Access Controls

A security policy explicitly defines the required security that a system enforces. These requirements are translated into a set of well-defined rules to determine whether a subject can access a specific object (Anderson, 1996). Access control models facilitate this authorisation and employ security policies. RBAC models have been widely used to capture dynamic requirements and execute the principle of least privilege (Joshi et al., 2001). RBACs are policy neutral and therefore provide flexibility in their design to allow extensions and enhancements, and are thus suitable for the integration of policies (Takabi et al., 2010; Le and Wang, 2012), which is important in the context of DIS.

Both security policies and access controls are used for collaboration and information sharing. Studies on integrating data for collaboration in different computing contexts, such as ubiquitous or distributed environments, have focused on security policy integration and conflict reconciliation (Begum et al., 2010; Kuang and Ibrahim, 2009; Yau and Yin, 2008). Several approaches are aimed at achieving policy integration and resolving conflicts between security policies. In ubiquitous computing, each organisation has its own security policy and ensures that its own security policy is well-maintained and no

authorised access is granted unless it satisfies its policy (Begum et al., 2010). In Software as a Service (SaaS) applications that collaborate, data leakage needs to be mitigatied to protect the shared sensitive and personal data. Wang and Jin (2011) proposed such an access control model, which works in SaaS applications to provide defence in-depth against data leakage. The mechanism has three steps: each organisation encodes its rules as MAC policies, an attribute-based recommender is used to suggest and prioritise recipients of the data, and abnormal recipients are examined by the system to provide a last line of defence.

To ensure the security policies of the entities participating in a DIS continue to be preserved, a global policy can be created to combine systems policies. Duan et al. (2016) proposed an automated policy combination to secure data in cross-organisation collaboration. Basically, the attribute rules in each policy are classified, then the rules limited to each attribute are reduced, and finally those rules are combined in a global policy. To continue policy preservation, the DIS security policy needs to be integrated with the global policy as well as their interaction managed (Yau and Yin, 2008). However, having a single authority to manage the global policies (Begum et al., 2010) is also a challenge.

Haddad et al. (2012) proposed a policy integration method that combines the authorisation policies of data sources integrating and sharing data. The global policy is enforced within the mediator. A user query is analysed against an integrated global security policy of the participating data sources. If the user has the right to access the data source, a result is returned; otherwise the query will be rejected. One of the limitations of this approach is that, while it covers the security policies generated by data sources and the integration location, it does not consider the actual data integration approach and the data consumers. Neither does it consider how much trust is afforded the entities that provide datasets.

In healthcare, where privacy is a significant concern, several approaches have been proposed to enforce privacy through access control models that can be used in e-healthcare services, such as Hung (2005). Li et al. (2009) proposed an access control model that uses a generalisation boundary technique which provides privacy and usability. It controls the amount of information exposed rather than just allowing information according to user roles. Kuang and Ibrahim (2009) also addressed privacy with security policy integration and access controls. Lamb et al. (2006) proposed an RBAC that manages the data service integration in health domains, where the consumers' queries are analysed against the access control policies.

2.4.5 Formal Analysis of Security Policies

This area focuses on automated methods and tools to detect and correct errors in software and security policy specifications before they are deployed. One example from

industry is the Amazon web services (Newcombe et al., 2015). Their experience with formal methods was useful in designing complex systems including cloud services to find errors that could not be found in other techniques.

Proposals for the analysis of security policies mainly differ in the formalism and tools used for policy modelling and analysis. These approaches pursue techniques ranging from SMT[27] formulae to Multi-Terminal Binary Decision Diagrams (MTBDD) and different kinds of logics.

Margrave (Fisler et al., 2005) uses MTBDDs as the underlying representation of XACML policies. It supports two types of policy analysis: policy querying, which analyses access requests evaluated to a certain decision, and change-impact analysis, which is used to compare policies. However, BDD-based approaches allow the analysis of policies only against a limited range of properties.

Alternative approaches encode policies and properties as propositional formulas and analyse them using SAT solvers (Hughes and Bultan, 2008). However, SAT solvers can only handle Boolean variables and therefore are limited in the type of access control policies that can be modelled and analysed.

Other formalisms have also been used. Description Logic (DL) (Kolovski et al., 2007) is used to formalise access control policies and employs off-the-shelf DL reasoners for policy analysis. The use of DL reasoners allows more expressive access control policies to be modelled but it suffers from scalability issues.

Answer Set Programming (ASP) (Ramli et al., 2013) does not support quantifiers, and does not easily allow the expression of constraints, such as Linear Arithmetic.

More recent approaches are based on SMT (Turkmen et al., 2015). The use of SMT not only enables a wider coverage of access control policies compared to the tools mentioned above, but also improves performance.

Other work has applied Event-B formal method (Abrial, 2010) to model and analyse access control policies (Hoang et al., 2009; Butler, 2013). The main advantage of using Event-B is that it is possible to model not only the access control policies but also the system where the policies are going to be deployed. Another advantage is the expressiveness of the Event-B formalism that allows modelling fine-grained policies. Lastly, the Rodin tool (Abrial et al., 2006), which supports the Event-B formalism, allows Java code generation from the formal model of the system and its policies.

2.5 Literature Critique and Research Gap

The scope of this research is discussed in the following points.

[27]SMT: Satisfiability Modulo Theories

1) Data Leakage Threats to DIS: There is limited discussion of these threats in the literature. The work mainly focuses on the violation of privacy on behalf of data consumers and similar entities. Violation of security as a property is dealt with separately by the literature, not necessarily in a DIS context. Likewise, trust is not well discussed in the DIS context as it was assumed in most studies in DIS.

2) Combination of the CPT Properties: Combining these properties in approaches that secure systems, not necessarily in DIS, has recently gained attention. Systems security is complicated and influenced by many factors and therefore it cannot be addressed in one way. Morton and Sasse (2012) proposed an integrated framework to create an effective privacy routine for any information system. Fléchais et al. (2013) considered human factors and took a holistic perspective in combining the CPT. He discussed authentication in the banking context of Saudi Arabia. Other studies, such as Manan et al. (2011), emphasised the need for combining CPT properties. A work on federated identity and access management created an access control solution that encompasses the CPT properties together (Khattak et al., 2012).

However, none of the approaches reviewed had focused on combining the CPT properties within the DIS security. Most studies have been carried out on DIS privacy through privacy-preserving approaches (Bhowmick et al., 2006; Yau and Yin, 2008; Pasierb et al., 2011) in data integration and data mining techniques. Some publications discuss security in the context of integration through the integration of security policies and access controls (Begum et al., 2010; Yau and Yin, 2008; Kuang and Ibrahim, 2009). For security requirements, a separate body of literature is concerned with combining the security polices of entities collaborating, and not particularly in the DIS context (Cruz et al., 2008).

Few studies have supported the idea that the integration should not only be privacy-preserving but also be based on security policies. This is where confidentiality, through access control, is combined with privacy in the integration process (Haddad et al., 2012). However, there is an evident lack of tackling trust among the properties covered.

Researchers have not treated trust in the context of the DIS in much detail. In any case, many approaches simply assume that the entities collaborating or integrating are trustworthy, especially when integrating data that originates from their own data sources. This perspective is briefly discussed in van den Braak et al. (2012) where security and privacy is achieved by using a trusted third party to manage access controls to private data. Since data is integrated within the organisation, it does not reflect the actual combination of the CPT properties.

3) Combining Data from Different Organisations: Integrating data within the organisation could be considered safe. The reason is that many of the entities involved in the integration process are subject to the same security measures. The data sources are

well-known and the integration location is within the organisation's boundary. Therefore, the concerns about security are controllable, to some extent.

However, integrating data outside the organisation can be problematical because many of the entities involved in the process use different security models and have different privacy requirements. These entities can be data providers or integration locations, which may not be secure enough. Integrating data coming from outside the organisation raises issues about security policies. Especially when integrating data outside the organisation's boundary, integrating security policies must include the organisation's own policies and the government legislation related to data protection, to ensure security of the overall system.

4) Investigating the Main DIS Components: From the perspective of securing the main components comprising DIS, the approaches presented here provide practical solutions expected to solve problems in a specific DIS component, such as the integration approach or the data sources side. However, the security of the whole system is hardly discussed in the literature, reviewed in Section 2.3.3.

5) Software Engineering Perspective: The studies on securing DIS found an architecture, a method, or an access control as explained, but lack a software engineering perspective. This means that the approaches are either at a high architectural level, which does not provide details for a software engineer to employ, or they are a precise algorithm that does not consider the picture of the whole system. The literature lacks the combination of these perspectives, which is what this study aims to achieve.

Based on the previous elements, a systematic search found only a few papers that are closely related to this study, see Section 2.3.3. Table 2.3 compares between the studies based on the elements of the scope.

Table 2.3: Comparison between Approaches to Secure DIS based on the Elements of the Scope of this Study

	Clifton et al. (2004)	**Bhowmick et al. (2006)**	**Haddad et al. (2012)**	**van den Braak et al. (2012)**
(1)	Privacy violations	Privacy violations	Privacy and Confidentiality violations	None
(2)	CP	CP	CP	CP
(3)	✓	✓	Not clear	No, within government organisations
(4)	✓	✓	✓	✓
(5)	No	✓	✓	No

2.6 Summary

This chapter started with the background of DIS and the threat of data leakage in those systems. This was followed by a discussion of approaches to secure DIS. As a result of reviewing these approaches, the existing literature fails to:

- Focus on data leakage as a threat to DIS that affects the confidentiality and privacy of the data.
- Consider the combination of the CPT properties in one approach to protect against data leakage.
- Cover the DIS that integrates data sources submitted from different organisations that have different security and privacy requirements.
- Provide a holistic perspective of the basic components of the DIS when investigating security and privacy.
- Provide clear guidance on how to mitigate the threats of data leakage in a way that assists software engineers.

This study aims to address these deficiencies by applying the concept of building secure systems by design. The chapter presented a combination of tools to achieve that, such as the use of threat modelling, security guidelines and polices, and the use of formal methods to evaluate security policies. The literature has been examined in terms of the scope of this study. The only works that were considered relevant are shown in Table 2.3, which establish the need of this study. The next chapter presents the research methodology adopted to reach its objectives.

Chapter 3

Research Methodology

This chapter identifies the research questions that this thesis aims to answer and discusses them. The research methods employed to address the research questions are then explained. Following this is a brief description of the research approaches used in software engineering and information security that have been selected to evaluate this research's contributions. Informed by these different methods, the chapter explains the research methodology used.

3.1 Research Questions

When DIS combine sensitive data across organisations that have heterogeneous security and privacy requirements, these requirements need to be maintained throughout the integration process to protect the data. These systems are susceptible to data leakage, as explained in Sections 1.1 and 2.2.

The gap in studies that focus on mitigating data leakage threats in DIS considering the properties of Confidentiality, Privacy, and Trust (CPT) by design, requires a novel approach (see Section 2.5). Consequently, the research questions presented here generally aim to propose, evaluate, and apply an approach to mitigate data leakage threats in DIS that software engineers can benefit from.

It is important to incorporate security into the early stages of system development, so that security goals can be achieved earlier in the cycle and threats can be controlled, see Section 2.4. A threat analysis can be used to elicit the data leakage threats in DIS that the system should be protected against. Therefore, the first research question is:

RQ1: What data leakage threats affect personal and sensitive data in DIS, focusing on the CPT properties?

The outcomes of the research question RQ1 are a conceptual DIS architecture, confirmed to represent DIS with a middle layer, and a list of carefully studied data leakage threats possible in that architecture.

To help software engineers introduce security from early stages of development, a mitigation approach, namely SecureDIS, is proposed as the main contribution of this study. SecureDIS starts with the basic architectural components resulting from research question RQ1. In addition, it contains a set of design guidelines directed to software engineers that assist in mitigating the threats elicited from RQ1, where each guideline is linked to one or more threats. The guidelines are confirmed by experts in the field in order to use them in practice.

To demonstrate the practical use of SecureDIS by software engineers, a formalism is employed. The formalism involves modelling a set of security policies that capture the CPT properties included in the guidelines. The model covers the core interaction between the data consumers and the data sources in DIS. The purpose of this activity is not only to utilise SecureDIS, but also to demonstrate the correctness and consistency of the model in capturing the properties, using formal methods. Hence, the second research question is:

RQ2: How can software engineers mitigate data leakage threats during the design of DIS?

To ensure that SecureDIS can actually be applied in practice, and to gather feedback from software engineers on its strengths and shortcomings, the study of a real data integration project is essential. The project selected needs to be within the scope of SecureDIS to assess how applicable SecureDIS is to it. In addition, the project must be amenable to being understood by employing several data collection methods. Therefore, the final research question is:

RQ3: To what extent is the proposed mitigation approach against data leakage applicable to a real data integration project?

The outcomes of research question RQ3 is the degree of applicability of SecureDIS to the project and the extent of it being accepted by software engineers working on that project.

3.2 Selected Research Methods

A mixed method of both qualitative and quantitative approaches is considered most suitable for the research questions. The following sections provide an overview of each

research method. Table 3.1 summarises the methods/activities and the research questions they aim to address, and their outcomes.

Table 3.1: Summary of Research Questions, Selected Research Methods, and Research Outcomes

	Part of the Question	**Research Method or Activity**	**Outcome**
RQ1	The DIS	Expert reviews (1) with open-ended questions	A confirmed conceptualised DIS architecture and data leakage locations.
	Data leakage threats	Threat analysis	A list of data leakage threats in DIS focusing on the CPT properties.
RQ2	A mitigation approach against data leakage	Expert reviews (2) with a questionnaire	A confirmed SecureDIS framework and guidelines.
	The use of the approach by software engineers	Event-B formal method	SecureDIS implementability by software engineers.
RQ3	Applicability of the mitigation approach	A case study with 3 methods:	SecureDIS applicability to a real project.
		1. Interviews	Exploring the data integration project.
		2. Questionnaires	Assessing the compliance to SecureDIS.
		3. Focus groups	Confirming the applicability findings.

3.2.1 Quantitative Research

Quantitative research collects numerical data to define, predict, and/or control a phenomenon (Gay, 1996). It aims to make a valid description of a phenomenon and show how it is controlled by variables with a minimum contact with the subject of the research, and therefore limiting personal bias on the analysis and interpretation of the data (Taylor, 2005). It uses fixed instruments that contain closed questions, which can be surveys and experiments (Creswell, 2003).

Quantitative research is evaluated by either descriptive or inferential statistics (Taylor, 2005). Descriptive statistics are used to describe the characteristics of a specific sample of data, while inferential statistics are used to determine the likelihood of generalising the characteristics from small samples to larger ones (Taylor, 2005).

In information security, quantitative approaches have been employed for measurement and evaluation. For example, Islam and Falcarin (2011) utilise questionnaires to check whether a security goal is achievable. The results are used to build security metrics

with the help of the Goal Question Metric (GQM) approach introduced by Basili et al. (1994).

The quantitative research methods employed in this study are limited to *questionnaires*, also part of the qualitative methods, as the aim was not to achieve statistical significance but to ask the participants the same questions with a scale. Those questionnaires were used during the expert reviews 1 and 2, to address research questions RQ1 and RQ2. To obtain *quantitative* data, experts can be surveyed, using questionnaires, to obtain their perspective on the item presented. The aim of questionnaires is to answer research questions related to "*how much*" or "*how many*" (Yin, 1984). The questionnaires thus need to be well-designed, follow the research aim, provide clear questions, define proper scales of measurement, and enable a suitable data analysis method to be chosen.

The analysis of the questionnaires follows the quantitative research approaches, where statistical analysis, if required, derives the results needed to reach the research aim. In addition, questionnaires were part of the focus group in research question RQ3 to assess the usefulness of the guidelines. A Likert scale (Likert, 1932) was used in all the questionnaires to provide suitable choices to the participants. They were self-administered (ching Leung, 2001) to suit the type of participants.

3.2.2 Qualitative Research

To interpret a phenomenon in its natural setting and to make sense of it, qualitative research methods are used (Trumbull, 2005). *Qualitative research* aims to obtain an insight into a phenomenon by collecting descriptive data on many variables over time (Taylor, 2005; Gay, 1996). Qualitative methods aim to answer a question of "*what*" and "*how*" (Yin, 1984), and they can be interviews, observations, documents, open-ended questions, and audio-visual data (Creswell, 2003; Taylor, 2005). To derive results and answer research questions, the analysis of texts and images (Creswell, 2003) could also be used. In software engineering and information security, case studies can be a qualitative method. He et al. (2006) is an example of using case studies to confirm requirement-based access control policies.

Three qualitative research methods are employed in this study, as follows:

Expert reviews are a guided one-to-one interview with experts in the field. Reviews are considered an informal validation method, as they are based on human subjectivity, but that does not imply they lack structure or do not use techniques (Balci, 1994). The reviews are a qualitative research method often used in computing (Holz et al., 2006). They can be used to evaluate a specific element (Balci, 1994). This element can be, for example, a user interface (Dumas et al., 1995), where experts can play the role of inexperienced users to identify usability problems (Holz et al., 2006).

The data resulting from these reviews can be both qualitative and quantitative, depending on the material presented to the experts, and so can be considered as a mixed method approach. To obtain *qualitative* data, experts are interviewed and asked open-ended questions. The interviews and the analysis of the data need to follow a qualitative research approach. Expert reviews were employed in research questions RQ1 and RQ2 to obtain feedback on the DIS architecture and the SecureDIS guidelines.

Interviews are one of the common qualitative data collection methods, and they can be structured, semi-structured, and unstructured (Gill et al., 2008). As part of research question RQ3, semi-structured interviews were employed. This type of interview provides flexibility to both the interviewer and the interviewee to respond to questions with more detail (Gill et al., 2008), which is needed in for the case study. The questions within the semi-structured interviews elicit expected information alongside other unanticipated information (Seaman, 1999).

Focus groups are similar to less structured interviews; however, they provide a guided group discussion to achieve research aims (Gill et al., 2008). Focus groups are employed in research question RQ3 to seek feedback from participants (Gill et al., 2008) on the analysis of the data collected by the case study methods.

To analyse qualitative data, the discussions should be transcribed and a thematic analysis, either inductive or deductive, is employed (Braun and Clarke, 2006). NVivo is one of the available tools to assist in coding such transcriptions. NVivo requires defining nodes (or codes) to demonstrate links between qualitative data. To demonstrate rigour in this process, a hybrid approach was used in this work that combined data-driven thematic analysis (Boyatzis, 1998; Braun and Clarke, 2006) and a deductive coding approach (Fereday and Muir-Cochrane, 2006).

3.2.3 Mixed Method Research

Mixed Methods research involves using both quantitative and qualitative methods in a single study (Lister, 2005) to collect data and answer research questions. It aims to combine the advantages of both approaches. It utilises open-ended and closed questions to collect data (Creswell, 2003), and therefore combines both statistical and text analysis (Creswell, 2003). An example of the use of mixed methods is assessing risk in information systems, which usually involves a combination of qualitative and quantitative data (Aagedal et al., 2002) to obtain the best results. Case studies are a good example of mixed methods if they combine qualitative and quantitative methods, as explained in Section 3.2.2.

A case study is the investigation of the uniqueness and complexity of a single interesting case to understand the details of its interactions with its context (Stake, 1995). This

method is used to study a contemporary phenomenon in a real-life context (Yin, 1984). Case studies aim to answer the questions "*how*" and "*why*" (Yin, 1984). They can be explanatory, exploratory, or descriptive, depending on the research question (Yin, 1984).

Data gathered within a case study are normally qualitative; however, quantitative data can also be gathered. Data can be collected by different methods, such as surveys, interviews, observations, or document reviews. Case studies can be conducted on a single case or multiple cases. Single case studies thoroughly examine a single organisation, group, or system, while multiple case studies examine a number of organisations or contexts (Holz et al., 2006).

Yin (1984) argues that case studies are often viewed with prejudice due to their perceived lack of rigour, difficulty in generalising the results, and the extensive time needed to conduct them. In addition, they produce many results that are hard to interpret. Yin states that case studies need to be well-designed to overcome these common flaws, and suggests five components of good design: a study question, its propositions, if any, its units of analysis, the logic linking the data to the propositions, and the criteria for interpreting the findings.

A case study is employed here in research question RQ3 covering three methods: interviews, questionnaires, and focus groups. This follows the triangulation technique that uses more than one source to collect data (Yin, 1984), which provides more accurate conclusions. The triangulation can also be methodological, as the data collection methods are of different types: qualitative and quantitative (Stake, 1995). The data gathered are analysed by linking SecureDIS to the project covered, as shown in Figure 3.1. Conducting and reporting the case study in this reseach followed the approach of Runeson and Höst (2009), which discusses case studies in software engineering.

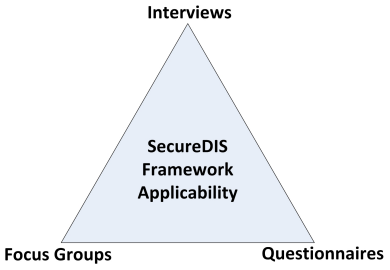

Figure 3.1: Data Collection Methods Triangulation in the Case Study

3.3 Research Methods Related to Discipline

Holz et al. (2006) claim that research methods usually depend on the computing discipline, as many computer science publications address research methods in specialisms. Therefore, the research methods employed here need to be related to the fields of software engineering and information security disciplines. The methods already discussed

were tailored to the discipline, but some methods are particularly employed here. The following sections explain the methods used in both software engineering and information security.

3.3.1 Software Engineering

Software engineering, unlike well-established areas of research, lacks guidance in terms of research paradigms (Shaw, 2002). Hence, Shaw (2002) proposed an approach to characterise research in the fields of science and engineering. In the proposed approach, Shaw identified a number of aspects, described in Table 3.2.

Table 3.2: Characterising Research in Software Engineering (Shaw, 2002)

Aspects to Identify	**Examples**
Types of question	o Methods or means of development o Method for analysis o Design, evaluation, or analysis of a particular instance o Generalisation or characterisation o Feasibility
Types of results	o Procedure or technique o Qualitative or descriptive model o Empirical model o Analytic model o Notation or tool o Specific solution o Answer or judgement o Report
Types of validation	o Analysis of the results such as formal analysis, empirical model, and controlled experiment. o Experience of use of results by other people to find them correct, useful, or effective. o Showing an example of how the results work. o Evaluation of the results against a given criterion. o Persuasion of the results by describing related techniques, systems, models and a working system for feasibility research questions. o Blatant assertion where no serious attempt is made to evaluate the result.

Reflecting on Shaw's proposal, research questions RQ1 and RQ2 are considered as a "*design, evaluation, or analysis of a particular instance*," as the questions aim to analyse the DIS to find a secure design for it. The results of both questions fall under the "*qualitative or descriptive model*," as the main contribution is a qualitative framework.

Research questions RQ2 and RQ3 aim to validate the proposed framework and its guidelines. The two questions are considered as the "*Experience of use of results by other people.*" The use of formal methods allows the extent to which the guidelines are implementable by a software engineer to be demonstrated. The case study evaluates the guidelines' applicability in a real data integration project by project team members. Research question RQ2, that evaluates the framework using formal methods, can also be considered as the "*Analysis of the results such as formal analysis, empirical model, and controlled experiment.*"

3.3.2 Information Security

Information security is concerned with the protection of information as an asset (von Solms and van Niekerk, 2013). In a panel discussion on cyber science, Maxion et al. (2010) categorised research in the field of cyber security to be: theoretical based on mathematics or logical syllogisms; or experimental that is not purely scientific. The authors had a few interesting questions regarding the effective use of scientific methods in the security field, and raised a concern whether security is a scientific discipline or a category of engineering technology.

Brostoff and Sasse (2001) argued that the goals and issues of the security field are similar to those of safety-critical systems. One reason for this similarity is that both fields provide the secondary goal of a system, which is to protect an entity and its staff while achieving the first goal of the system. However, they identify the main difference between the two is the fact that violating security can be adversarial, while the violation of safety is caused by unintentional system failure.

In terms of the research methods employed in the field of information security, Siponen and Oinas-kukkonen (2007) surveyed the literature on security issues and the approaches to address them: access to information systems, secure communication, security management, and development of secure information systems. They argued that most research in information security focuses on the technical context in addition to mathematical approaches with a philosophical logic. They believe that the use of empirical studies based on theories in other disciplines is crucial. One of the ways to achieve that is through utilising qualitative and quantitative methods.

Butler (2002) argued that empirical research in security is challenging as eliciting research data is hindered by limited access to people and data in organisations, in addition to the management perception that the research benefits do not exceed the cost of conducting the research. Thus, to conduct research in security, Fléchais (2005) proposed the use of social-science research methodologies to develop and evaluate security design methods. Moreover, threat analysis and risk assessment can be used as perspectives

that guide a security approach. Section 2.4.2 discusses the importance of threat analysis in building secure systems.

3.3.3 Formal Methods

In software engineering, one of the ways of constructing reliable systems despite their complexity is by the use of formal methods (Clarke et al., 1996). Formal methods have been widely used to specify systems rigorously. Systems built on mathematical elements, such as sets or functions, enable the analysis of these elements to achieve accurate properties, such as completeness and consistency (Butler et al., 2004).

Verification using formal methods is achieved in two ways: model checking and theorem proving. Model checking is an automatic process that provides useful information about a system's correctness by building a finite model of the system, while theorem proving uses the system's axioms to prove a property (Clarke et al., 1996).

Event-B is a formal method that can be used to model systems for specification and verification. Event-B was extended from the B-Method (Abrial, 2010) and is a state-based method that uses set theory as a distinctive attribute (Butler, 2013). A system can be modelled gradually to reflect its complexity by the use of abstraction and refinements. The mathematical proofs in Event-B are used to ensure the correctness of each level and the consistency among all modelled levels (Butler, 2013).

In the area of computer security, formal methods provide a structured approach for modelling systems using accurate notations (Butler et al., 2004). The notations can capture the security policies, system properties, and underlying assumptions. Formal methods are used as a verification tool to provide assurance that the system built meets its security goals, and as a specification tool to ensure the system's design is captured (Wing, 1998).

3.4 Research Methodology

The selected research methods in Table 3.1 form the research methodology used here. The methodology consists of three phases, shown in Figure 3.2.

The ***First Phase*** is concerned with the threat analysis. This starts with a literature review and understanding of the research problem, see Chapter 2. The review results in a conceptualised DIS architecture and identification of the leakage locations that are then assessed by the first expert reviews. The threat analysis is conducted on the confirmed architecture to elicit data leakage threats covered by this study and guided by the CPT properties, see Chapter 4.

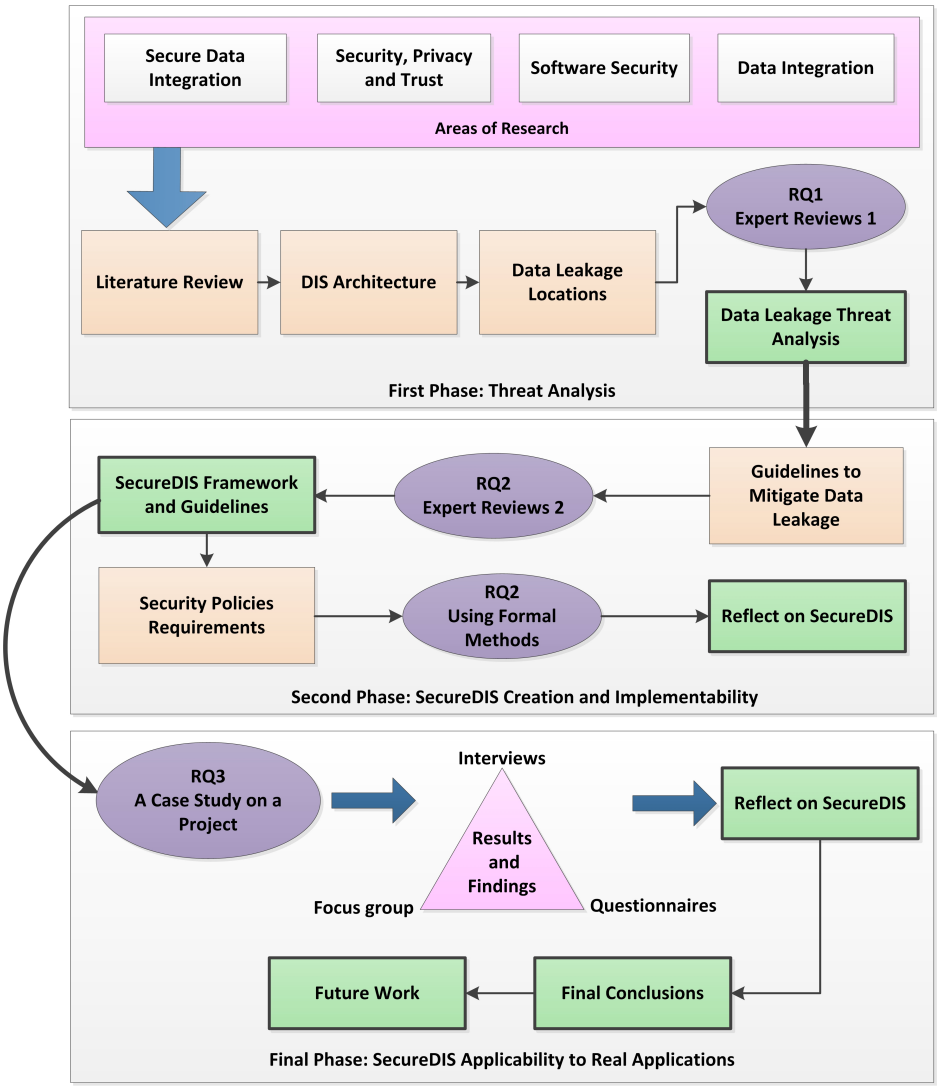

Figure 3.2: Research Methodology

The ***Second Phase*** addresses the needs of software engineers by creating the SecureDIS framework using the components from the first phase and proposing mitigation guidelines to overcome the threats resulting from the threat analysis. To confirm the validity of the proposed guidelines, a second expert review is conducted, see Chapter 5. Whether the SecureDIS guidelines are implementable by software engineers is assessed by creating DIS security policies using the Event-B formal method, see Chapter 6.

The ***Final Phase*** assesses whether SecureDIS is applicable to a real data integration project by conducting a case study, see Chapter 7. The result of this phase is conclusions about the degree of applicability of SecureDIS, see Chapter 8.

3.5 Summary

This chapter has discussed the research questions. It has also discussed the research methods relevant to the fields of software engineering and information security, which would be suitable for answering the research questions, see Table 3.1.

The main contribution targets software engineers by proposing a novel data leakage mitigation framework with a set of guidelines, namely SecureDIS. SecureDIS is based on a conceptualised DIS architecture and a threat analysis to elicit data leakage threats. Expert reviews were proposed to confirm the validity of the architecture and data leakage locations within it. The guidelines, within SecureDIS, are to be confirmed by security experts using a mixed methods approach. For a qualitative assessment of the guidelines, a second set of expert reviews will be conducted. However, for a quantitative assessment, the expert reviews include a questionnaire that aims to statistically accept or reject the guidelines. To assess the implementability of the guidelines by software engineers, the guidelines will be formalised and modelled to create security policies using Event-B formal method.

Once the SecureDIS framework and guidelines are confirmed, applicability assessment is conducted with a case study on a real data integration project. The research stages and the employed methods alongside the research questions were illustrated in the research methodology, see Figure 3.2.

Chapter 4

Data Leakage Threat Analysis in DIS

This chapter presents a process to analyse data leakage threats in DIS. The analysis starts by building a conceptual DIS architecture with a middle layer that represents the focus for the analysis. The architecture is a used to indicate the locations of possible data leakage. Then follows a discussion of how security experts have reviewed, validated, and confirmed the proposed architecture. This discussion also covered how data integration is implemented in practice, which is useful when proposing suitable approaches to mitigation.

In the light of the properties of Confidentiality, Privacy, and Trust (CPT), the process specifies the assets, investigates leakage threats, and provides findings of the analysis. The threat analysis process addresses the research question RQ1 defined in Section 3.1.

4.1 The Threat Analysis Process

DIS with middle layers are susceptible to data leakage as discussed in Section 1.1 and Section 2.2.3. To determine potential vulnerabilities in the DIS that may cause data leakage, a threat analysis is conducted. Identifying these vulnerabilities helps in designing a mitigation approach so that secure systems may be built.

Some of the existing threat analyses, such as STRIDE and LINDDUN, see Section 2.4.2, suggest using a detailed Data Flow Diagram (DFD). The elements of the DFD are evaluated against security threats in the case of STRIDE and against privacy properties in the case of LINDDUN. The threat analysis in this chapter is not conducted on a complete and running system; hence, a detailed DFD of the system activities cannot be created. Therefore, the proposed analysis is based on a conceptualised architecture of the DIS that consists of the components essential to achieve data integration. The DIS

components' locations within the architecture are used as a guide to elicit data leakage threats. The proposed threat analysis process is detailed below.

- **Step 0:** Understand the DIS and the data leakage problem. The result is a confirmed conceptualised DIS architecture and data leakage locations that are used in the following steps.
- **Step 1:** Identify the assets that the system aims to protect.
- **Step 2:** Identify the security properties. These properties are used as a guide for the analysis alongside the architectural components of the DIS.
- **Step 3:** Identify threats that violate the security properties. Based on leakage locations within the DIS, the violations of the CPT properties lead to identifying the threats.
- **Step 4:** Analyse and present the findings.

Threat analysis approaches usually include recommended countermeasures against the threats identified. However, in this work, the countermeasures will be delayed until a complete picture of the system is reached. The following subsections explain the details of each step of the analysis.

4.2 Step 0: Understand the DIS and the Data Leakage Problem

The following subsections explain the conceptualised DIS architecture and how it was used to identify data leakage locations. This is followed by how the experts' reviews were conducted to improve the proposed architecture and help in understanding the data integration employed in practice.

4.2.1 The Conceptualised Architecture of DIS

An architecture is proposed that captures the functionality and components to be used as the basis for this study. The architecture is based on several DIS, discussed in Section 2.1.3, and includes the following components:

1. **Data and data sources:** the core components of the DIS. They represent the sources integrated to answer data consumers' queries, see Section 2.1.1.
2. **Integration approach:** the method used to integrate the data, see Section 2.1.2.

3. **Integration location:** where the integration process takes place to answer data consumers' queries, see Section 2.1.3. The integration location is the place where: 1) data coming from different sources is analysed, processed, and integrated following the integration approach, 2) the security and privacy requirements of the data sources are received, and 3) the consumers' queries are received and appropriate transformations are conducted.

4. **Data consumers:** the client side of the system. Data integration is usually requested by data consumers who send queries to the integration location to provide consumers with the results, see Section 2.1.1.

The architecture has three layers, the first of which is the data consumers' layer, where consumers' queries are created and results are returned. The second layer (the middle layer) processes queries, and it contains the integration location and the integration approach. The third layer contains the data and data sources. Figure 4.1 illustrates how the components interact with each other when a consumer requests a query.

This architecture assumes that the integration process occurs at the middle layer rather than on the data sources' side or on the data consumers' side. It also assumes that the integration process may use external entities to assist in the integration process, such as third parties and cloud services. Finally, it assumes that data sources participating in the integration process come from different organisations and hence have different security and privacy requirements over sensitive data.

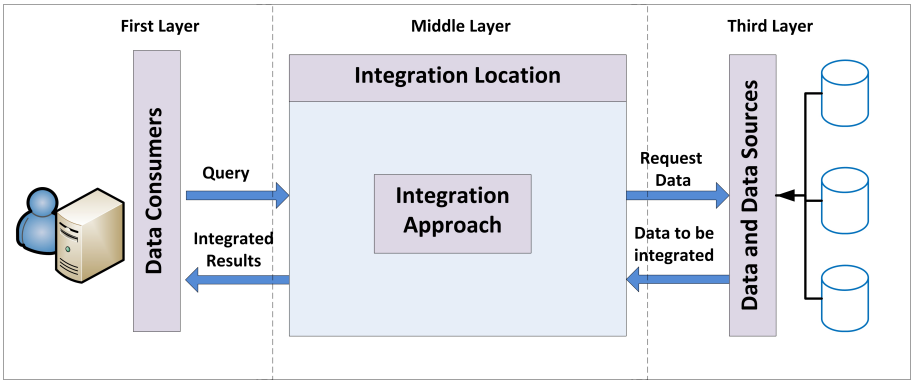

Figure 4.1: Conceptualised DIS Architecture with a Middle Layer

4.2.2 Identifying Data Leakage Locations

As shown in Figure 4.1, there are three possibilities for data leakage in the architecture: leakage from the data consumers' side; leakage from the middle layer; and leakage from the data sources' side. Figure 4.2 shows these leakage locations within the architecture. The leakage locations are discussed below.

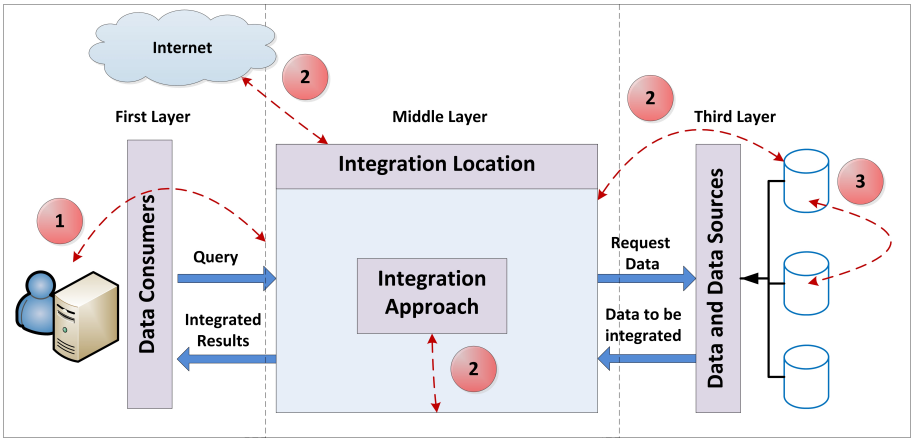

Figure 4.2: DIS Architecture with Leakage Locations

1. **Leakage from the data consumers' side:** Many security attacks occur in this layer, shown as leakage 1 in Figure 4.2. This layer is where access to data is requested and returned. Hence, the security and privacy policies of the system in general, and the data sources' in particular, are enforced in this part of the system. The risks associated with consumers' requests to sensitive data need to be considered in this layer. Therefore, it is important to include this location in the threat analysis.

2. **Leakage from the middle layer:** The middle layer consists of the integration location and the integration approach. Data integration processing phases can be achieved in one or several layers. Data leakage in this layer can occur between the layers or between the layers and other entities, such as the internet or data sources, shown as leakage 2 in Figure 4.2. In addition, external entities, such as third parties or cloud services, can be used to assist the integration process, where their trustworthiness to protect sensitive data is unknown. Therefore, the threat analysis needs to investigate threats occurring in this layer.

3. **Leakage from the data sources' side:** The heterogeneity among the combined data sources in DIS is not only in the data models used. It can extend to other dimensions especially when data is provided by different organisations who may differ in their security models, privacy requirements, licensing, and regulations. Each data provider has its own level of trustworthiness in providing the correct data and consequently the security information about it. Failing to capture or enforce security and privacy policies and requirements of the data sources causes data leakage. Leakage can also occur between data sources, shown as leakage 3 in Figure 4.2, where inference attacks between sources are possible. Hence, this layer is included in the threat analysis.

4.2.3 Experts Evaluation

To proceed with the threat analysis, expert reviews were chosen as a method to confirm the validity of the conceptualised DIS architecture as well as the anticipated data leakage locations identified in Sections 4.2.1 and 4.2.2. The reviews aim to achieve the following objectives:

- **O1**: Confirm and discuss the conceptualised architectural components of DIS.
- **O2**: Determine how data integration is employed in practice.
- **O3**: Discuss data leakage and confirm the leakage locations within the architecture.
- **O4**: Discuss possible data leakage prevention approaches.

The following subsections explain the details of expert reviews carried out for these purposes (Section 3.2.2 explains expert reviews as a research method).

4.2.3.1 Participants Selection Criteria

To be included, a participant needed to have a sufficient understanding of security and privacy requirements in information systems, with significant experience in developing, consulting, or managing security for five years or more. It was also important to know the expert's role in a data integration setting. The experts were a combination of researchers and employees working in corporate and government organisations, in both Saudi Arabia and the UK.

The number of experts needed for the reviews is between three and five according to the findings of Nielsen and Mack (1994) in the heuristic evaluations of usability. Also, the number of experts depends on the degree of saturation reached during the reviews (Marshall et al., 2013). Although it was difficult to find experts in the area, and they have limited time to give for this activity, ten experts were contacted initially. Five of these agreed to participate, which is valid in terms of the number of experts needed in terms of both quantity and quality, Table 4.1 shows their details.

4.2.3.2 Material Design

A document was presented to the experts containing the conceptualised DIS architecture and a short description of its components with a scenario of a data leakage. It also included possible data leakage locations from Section 4.2.2 (see Appendix A for the material). The document included several open-ended questions linked to the review's objectives shown in Table 4.2.

Table 4.1: Experts Selected to Review DIS Architecture

Job Title	Organisation	Country	Expert ID
Data Management Consultant	Semi-government agency	SA	Expert A
Security Consultant	Private security consultation company	SA	Expert B
Security Chief Officer	National health services	SA	Expert C
Security Consultant	Government organisation	SA	Expert D
Researcher in Security	Higher education institute	UK	Expert E

Table 4.2: Questions Asked to Experts

No.	Question	Objective
1	From your experience, are the DIS architecture and components reasonable?	O1, O2
2	Does the scenario reflect real-life applications?	O1, O2
3	Do you have any recommendations to improve the architecture and its components?	O1
4	If you have a real-life application that does something similar to what is presented in the scenario, can you provide further details of it for citation in this research?	O2
5	Are the leakage locations provided in the scenario reasonable? If not, which ones are not reasonable and why?	O3
6	What data leakage prevention mechanism, in your opinion, can be applied at each data leakage location?	O4
7	Assuming that a *framework* provides a general customisable solution to this sort of problem, a *model* provides a solution to some aspect of the framework, showing how factors are related, and an *algorithm* provides a detailed working solution to a specific part of the model: what would you recommend as a proper practical and useful solution for data leakage (a framework, a model or an algorithm) and why?	O4

4.2.3.3 Review Procedure

Expert reviews lasted for four weeks from August to September 2013. An ethical approval was sought from the university with the number ERGO/FPSE/8911 to conduct this review. Four of the five participating experts were contacted by email to arrange for meetings to conduct reviews and one participant was approached in person. Upon agreement by signing consent forms, a MS Word file containing the content described in Section 4.2.3.2 was sent to all participants.

The data collected from the reviews was gathered as MS Word file attachments to email. Experts were reminded by email to complete the review. Experts completed the Word

document with their responses, and upon completion they were contacted by phone or in person for further discussion.

4.2.3.4 Review Analysis

Since there were so few experts and questions, there was no need to use computer analysis tools to analyse the qualitative data. The experts' responses were coded, analysed, and themed manually following the approach of Boyatzis (1998) for categorising qualitative data.

The process of qualitative analysis is shown below.

1. The discussions with experts were recorded, transcribed, and included with their responses, in the Word file. The voice recordings were discarded after use.
2. The transcriptions were anonymised, sensitive data being eliminated.
3. Experts' responses relevant to the objectives of the review were highlighted and grouped.
4. A set of codes related to the objectives of the review was created and linked to the experts' responses.
5. Another set of codes was created covering more of the experts' responses.
6. Both sets of codes were arranged in themes. The codes are listed in Table 4.3.
7. Experts' responses were listed under the appropriate themes.
8. Experts' responses that directly address the review objectives, were carefully examined and included in the final results.

Table 4.3: Codes Used to Analyse the Experts' Reviews

Review Objective	**Code**	**Theme**
O1	architecture, component	1
O2	data integration, applications	1,2,6
O3	threats, data leakage, location, layer	1,3
O4	mitigation, technique,	4
NA	security policies	1,5
NA	security by design, external data sources, human factors	6

4.2.4 Summary of Experts' Responses

The following themes summarise the experts' responses.

Theme 1: DIS Architecture and Components: Comments were made on the conceptualised architecture and suggestions for future improvements were proposed. In terms of the *architecture* and the *leakage scenario*, experts A, B, D and E thought they are reasonable and similar to many real-life applications. This was emphasised by expert A, who mentioned the existence of similar projects in their organisation that employ data integration in a similar fashion. However, expert C thought that the DIS components usually depend on the vendor and how the vendor implements data integration. Expert C also remarked that the materialised and virtualised approaches to data integration are considered types of data source rather than integration approaches.

In terms of *improvements to the architecture*, expert C suggested providing a standard format for the data sources component, such as the Configuration Management Databases (CMDB), to illustrate that data sources are unified. Expert C suggested adding more details regarding sharing the security protocols and parameters between the integration location component and the data sources component for visibility, and ability to exchange data. Expert C suggested adding a common set of data sharing and exchange policies to the integration approach component, to indicate each vendor's security policies and standards, which enables secure data integration. For example, a vendor might have a policy for sharing aggregates rather than complete data.

Theme 2: Data Integration in Practice: As many organisations employ data integration in their projects, experts were asked to provide examples of similar applications to help understand data integration from a practical point of view. Expert D knew of several data integration applications, but was not allowed to disclose these types of application for confidentiality reasons. However, expert B knew that there were tools that aim to mitigate data leakage, but they do not cover the whole system. Expert E did not know of any data integration applications at the time of the interview.

Expert B suggested several websites used in practice for security compliance and data protection, such as the RSA[1], Varonis[2] for insider threats, TITUS[3] to classify and share data, Digital Guardian[4] to prevent data loss, and network data leakage prevention products[5], in addition to several secure file transfer solutions, which are outside the scope of this study.

Expert C said that the real purpose of a DIS is the integration between several applications produced by the same vendor. Expert C discussed an example of a data

[1] https://www.rsa.com/en-us
[2] http://www.varonis.com/
[3] http://www.titus.com/
[4] https://digitalguardian.com/
[5] https://www.fidelissecurity.com/

integration application, namely the Security Information and Event Manager (SIEM), which integrates system logs and correlates them to create information used in updating the stakeholders about possible attacks.

Theme 3: Data Leakage Locations in DIS: Experts discussed the proposed data leakage locations and possible data leakage mitigation approaches. While expert D thought that the suggested locations were reasonable, experts A, B, and E discussed the locations in detail. The following is the summary of their feedback.

1. **Leakage from the data consumers' side:** Expert E suggested that providing more specific scenarios or case studies would be useful for this type of data leakage. Expert B suggested: 1) auditing access control rules and using identity management systems to find duplicate or inappropriate rules; 2) reviewing data access logs; and 3) using firewalls to prevent or limit multiple accesses to the same records. Expert A said, "*the use of row-level access control measures; SAP BO*[6]*, for example, ensures a user can access only the authorised rows (so it's finer than traditional solutions which offer table-level access). With powerful user management solutions and policies, the risk is minimised.*" However, expert A mentioned that, despite the masking techniques used with sensitive data, there is always the possibility of data leakage through inference and correlation of attributes.

2. **Leakage from the middle layer:** This leakage can be:

   - A leakage to data sources: Expert E thought that this leakage location is possible if the leakage is to an open access/public database, such as Wikipedia, or social media, such as Facebook. To ensure the integrity of the original data, expert B suggested using hashing and sandboxing techniques, in addition to ensuring that the resulting data is correct and has rules for access by humans or machines.

     Expert A defined this leakage location to be in two directions and suggested techniques to prevent data leakage from each direction as follows: "***[a) a leakage from data sources to DIS:]*** *deploy strong access control lists, utilise secure links, use firewalls with IDS/IPS*[7] *systems, and grant the media agent access to only a specific zone/environment (e.g. integration and pre-production).* ***[b) a leakage from DIS to data sources:]*** *use the security measures above. In addition, monitor and secure the zone from which external requests may be initiated.*"

   - A leakage to the internet or third parties: "*is a reasonable leakage*", according to expert E. However, to prevent it there is a need for data classification, access control, and identity management as expert B proposed. Expert

[6] SAP BO: SAP Business Objects, can be found here: https://www.sapbi.com/
[7] IDS/IPS: Intrusion Detection System / Intrusion Prevention System

A, on the other hand, suggested, *"never let the internet-facing architecture share individual records of data (only aggregated results, especially in Business Intelligence (BI) reports). If there was a need for sharing specific records, the sensitive fields in them are always masked with irreversible hashing approaches."* However, expert C was of the opinion that there are no firewalls that can go deeper to table/field level to prevent data leakage. Therefore, security solutions specific to databases need to be used to block particular fields.

- No feedback was given by the experts regarding leakage within the integration location layers.

3. **Leakage from the data sources' side:** Expert E thought that the link between data sources should be through the integration location component, and said, *"data sources can control only data access and they cannot carry out any actions independently."* To eliminate data leakage in this location, expert B suggested classifying the data and determining who is authorised to access it. Expert B stated that many tools avoid this type of leakage. Expert A suggested using encryption and secure certificates that use PGP PKI[8], to secure the data, on the grounds that it is the data providers' responsibility and therefore out of the integration scope.

Expert E explained that the legal ways to exchange information is through disclosure or data rights transfer to other entities, and anything beyond that is considered a data leakage. Regarding reasons for data leakage, expert B explained, "*we are talking here about the lifecycles of a data record/file. If we start with [the] classification of that record by who will have access [to] read/write/delete or archive, then we start correctly. If we wait until that record has been created, then add our requirements, then definitely we will have a data leakage problem!*" Expert E divided the causes of data leakage in a DIS to either a system, or a data consumer, exploiting the vulnerabilities of the DIS. In addition, expert E mentioned that data leakage threats can be accidental or deliberate and they usually occur in the higher three layers of the OSI Model[9].

Regarding the violated security properties causing data leakage, expert D suggested considering all security properties, such as authentication, access control, and Confidentiality, Integrity, and Availability (CIA), or at least the CIA with respect to the architecture.

Theme 4: Possible Data Leakage Mitigation Approaches: Experts A, B, and D suggested creating a framework to mitigate data leakage threats. Expert A thought a framework is suitable because it would enable "*truly understand[ing] what this research is proposing and what advances it is targeting.*" The reason for that, from expert's B

[8]PGP PKI: Pretty Good Privacy - Public Key Infrastructure: http://www.pgpi.org/doc/pgpintro/
[9]OSI: Open System Interconnection model, see (Zimmermann, 1980)

perspective is that "*it can accommodate human factors, machine operations, [and] tools that will be used by both.*" Expert B also recommended using the Mis-usability Weight Concept, known as the M-Score, to expand the algorithm to cover the whole life of the data record, and show the true importance of preventing data leakage. Expert D recommended a framework approach because "*it [would] give a general scope and many solutions can be extracted out of it. In the framework, all possible attributes (variables) should be determined and included to get a solid and secure solution.*" Expert E explains, "*a framework is a better option because it is flexible and scalable, a model is more restrictive.*" Expert E thinks that, in this case, an algorithm is not preferred, as there is "*no silver bullet in security, no one glove fits everybody.*"

From the perspective of expert C, data leakage mitigation depends on two elements: the data sharing and exchange policy used, and the access control model adopted. Expert C explained that data leakage prevention tools usually work in two phases. The first phase is categorising and marking the data as sensitive, secret, top secret, etc. The second phase is preventing data leakage from internal gateways by blocking outgoing data and/or end points, which involves blocking certain activities accomplished by computers, such as blocking emails, file uploads, use of USBs, etc. Expert C mentioned several times that "*everything is based on the policy*," and emphasised that data sources must be authenticated, registered, and validated through security checks, prior to use.

Theme 5: The Importance of Security Policies: The experts discussed how security policies must be considered as part of the approach to mitigating data leakage. In terms of using security policies to manage the data integration process, expert A's organisation uses two: the first is the data provider global policy that approves the masking techniques and decides which fields to be masked based on the sensitivity level of the data. The second policy is the organisation's local policy that deals with data security and privacy among all data integration projects carried out in the organisation. The latter is used as the security policy associated with the integrated data when requested by a client and is enforced by legal contracts.

Expert A mentioned one of the important aspects of the security policy, saying, "*security policies are usually reactive as opposed to proactive; when problems arise the security policy is created.*" Security policies are somewhat relevant to the existence of a data protection law. In Saudi Arabia, there is no specific data sharing and exchange law present at the moment. It is still being discussed by many organisations and there is much effort pushing toward having one. This fact was confirmed by experts A and C and in the work of Balouziyeh and Husein (2012).

Theme 6: Miscellaneous Aspects: One aspect that arose in discussion is securing systems by design. Expert A's organisation uses a layered architecture that limits the risk of direct attacks. They use multiple security policies, which can be helpful in covering most issues related to data security. Also, there is a real-time event-triggered

system monitor that records logs and gives security alerts. Expert B mentioned an important aspect of software security, which is incorporating security requirements at the beginning of the software development. Expert B emphasised this fact by saying, "*fixing applications for security is a nightmare after the fact. It must be included and approved by the end-users before a single line of code is written.*"

One of the experts discussed their organisation's practices in using data sources coming from different organisations in the integration process. Expert A's organisation receives data, in some cases, from a trusted third party that has access to sources provided by external data providers. In other cases, their organisation has direct access to data providers' sources. The data in both cases can be received through secure channels, such as Secure File Transfer Protocol (SFTP), and may be signed using digital certificates. In other cases, the data can be provided as physical CDs or email attachments.

Another aspect discussed was attitudes to security. Human factors were emphasised by expert B, as many tools may be used to enforce security policies but the human factor is important. Expert A shared this perspective, mentioning trusted third parties and the way they handle sensitive data, which is usually enforced by legal means. Expert A also emphasised the role of the organisation culture as a huge barrier to enforcing security policy and having clear security practices.

4.2.5 Limitations

Some experts were able to respond to questions in detail, others were cautious and limited by their organisation's privacy protection policy. The nature of answers to the open-ended questions varied between experts. It was very dependent on the experts' level of expertise and the nature of their job. To overcome that, the consensus among experts was sought.

4.2.6 Review Discussion

Whether there is true integration occurring in practice depends on the experts' area. Thus, from expert C's perspective, there is no true data integration occurring. Expert C was working with industrial vendors extensively and therefore provided vendor-oriented responses. However, expert A was conducting data integration in-house. Therefore, data integration projects and applications were very clear to expert A. This would explain the contradiction of responses between expert A and expert C. In addition, expert C suggested unifying the data sources to one common format, which is against the concept of integrating heterogeneous data sources. Hence, this indicates the differences among experts as to what data integration really is.

Data leakage locations were usually found in any web-based or distributed system. However, the unique aspect covered by this analysis is the data leakage occurring from combining heterogeneous data sources that vary in security and privacy requirements. Experts discussed general data leakage prevention approaches, such as adopting identity management, access control models, and firewalls that comply to the organisation's general security policy. However, most of the experts did not discuss data leakage in the context of DIS. This raises a need to create data leakage prevention approaches to accommodate the nature of DIS. Regarding the causes of data leakage, expert A touched briefly on the fact that data leakage occurs from inference attacks and attribute correlations, which were also found in the literature discussed in Section 2.2.3.1.

Mitigating data leakage in DIS can be achieved with several countermeasures. One important defence is the existence of a data sharing and exchange policy that covers security, privacy, and trust in DIS. Depending on the security policy adopted, leakage of a data aggregate may or may not be considered a data leakage, whilst leaking raw data may or may not be considered serious. For example, is it possible for pharmaceutical companies to be allowed to know the number of people who suffer a specific disease? The policy should state which data is allowed and how much of it can be exposed. Therefore, the existence of such policy is essential, as supported by experts A and C. The existence of a clear data sharing and exchange policy will influence the implementation of the access control rules, which can be tailored to prohibit specific fields from access by specific data consumers. In addition, the use of database firewall that monitors database traffic, such as the one by Oracle[10], can help to prevent data leakage, which was discussed by expert A and contradicts with expert C's opinion.

Data leakage mitigation through the use of security policies can be related to the data protection law implemented. In the case of Saudi Arabia, no comprehensive data privacy law is currently in effect, although a general government law covers some aspects of privacy, security policies become challenging. Therefore, companies use legal contracts, e.g. Service Level Agreement (SLA), to enforce their security policy on third parties and clients who consume integrated data. Authorised third parties may leak data to other entities or may illegally use transitive trusts to other entities. Therefore, ideally, there is a need to enforce technical mechanisms as well to label and track data distribution, such as watermarking or masking, to detect which third party did what. This issue was neglected by the experts as legal contracts covered their needs.

Another countermeasure to leakage is by considering the human factor. The possibility of an insider attacker is inevitable in any organisation and the possibility of intentional data leakage is no less risky than unintentional leakage. A recent report of data leakage occurring in 2015 by the InfoWatch Analytical Centre[11] indicates that 65.1% of the leaks

[10]Oracle's Audit Vault and Database Firewall, see https://www.oracle.com/database/security/audit-vault-database-firewall/index.html

[11]Can be accessed here, https://infowatch.com/analytics/reports/5184

in organisations are caused by internal violators. Therefore, in the DIS context, data consumers should be considered carefully. This can be achieved by applying security measures to limit their access to critical data to minimise the risk and prevent client side attacks. This is also applicable to people working within the DIS, such as system administrators and database managers.

4.2.7 Confirmed DIS Architecture and Data Leakage Locations

Based on the analysis of the experts' reviews, several changes were made to the conceptualised DIS architecture and leakage locations. The preliminary architecture, shown in Figure 4.2, did not include the security policies as a DIS component. However, based on discussions with experts A and C and the understanding of its importance as a countermeasure, supported by the work of Clifton et al. (2004), security policies were introduced as a new component. This illustrates its strong relationship with the way in which data is accessed, integrated and presented to the data consumers' component. Also, the data integration can be made more secure if the authorisation policies, i.e. security policies, are considered in the integration process, as suggested by Haddad et al. (2012).

The data leakage location within the data sources component, shown as location 3 in Figure 4.2, is changed to a clear line that passes through the integration location component. This is to emphasise the integration location component's contribution to such a leakage, as supported by expert E.

There is a need to manage how security is implemented in the DIS. Therefore, a System Security Management (SSM) component is introduced to the DIS architecture. The SSM should contain all the needed activities to implement, log, monitor, and audit security in the DIS.

Regarding the proposed approach to mitigate data leakage, experts A, B, and D supported the idea of implementing a framework that combines the essential techniques to ensure security and privacy. Based on their direct relevance to securing the data, the techniques were investigated and several ones were selected as part of the proposed framework. The chosen techniques are listed below.

- Using access control models (experts A, C, and E).
- Classifying the data (experts A, B, C, and E).
- Determining the entities that can access the data (expert B).
- Using encryption (expert A).
- Creating access rules for the integrated data returned by a query (expert B).
- Auditing access control rules (expert B).

- Reviewing access logs (expert B).

As a result of the reviews, the confirmed DIS architecture with data leakage locations is given in Figure 4.3. This architecture is used in the remaining steps of the threat analysis.

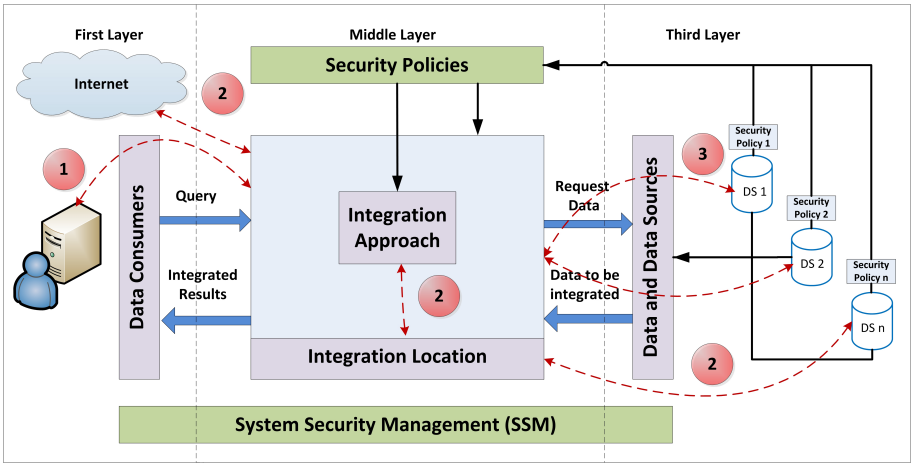

Figure 4.3: Confirmed DIS Architecture with Data Leakage Locations

4.3 Step 1: Identify the Assets

The first step is to determine the assets that the system aims to protect. This analysis is mainly concerned with protecting sensitive data. In this research, the term '*sensitive data*', based on Pearson (2009), includes the following:

- Data that directly identifies individuals, namely Personal Identifiable Information (PII) (Guarda and Zannone, 2009), such as name, SSN, etc.
- Data that indirectly identifies individuals, namely Quasi ID (QID) (Goryczka et al., 2013), such as address, birth date, gender, etc.
- Data that has value to the provider or the owner, where providers impose security and privacy requirements on it, such as trade secrets, financial data, etc.

Failing to protect *sensitive data* has many consequences, for example:

- *Identifying individuals*: by revealing their personal information and using their information for fraudulent purposes (ICO, 2017), or causing them harm.
- *Non-compliance by violating data protection laws*: exposing personal data causes law violations. This includes the general data protection legislations, such as the DPA in the UK or the legislation specific to a domain, such as HIPPA in the USA.

- *Non-compliance by violating policies*: sensitive data has restrictions on the amount of data used, retention period, purpose, etc. (Constante et al., 2013). Violating these policies, such as using the data against its intended purposes, or allowing data to be accessed by unauthorised entities affects the protection of sensitive data.
- *Information disclosure*: by exposing information that has consequences of loss to reputation, money, or rights. This includes exposing trade secrets that consequently cause financial harm, exposing sensitive harmful information about an entity that causes loss of its reputation, or exposing intellectual data that results in losing rights on the data.

4.4 Step 2: Identify the Security Properties

This step determines the security properties used as a guide to both elicit and mitigate data leakage threats. Many properties influence the unauthorised disclosure of personal and sensitive data. However, this analysis concentrates on the properties of CPT. The literature in Section 2.2.1 provides an overview of these properties. To manage the analysis, several elements are considered for each property, as follows.

Confidentiality: *limiting access to authorised entities* (ISO, 2014). Data leakage threats to confidentiality occur if, for example, the integration location returns the integrated data requested to a consumer who was not allowed to access the data, based on the data sources' security policies. In another example, leakage can occur if sensitive data is transferred as plain text between the DIS components.

Hence, the confidentiality elements covered by this analysis are:

- *Access Control*: governs the authorisation of entities to access sensitive data. For example, using the RBAC model and configuring it so that each user can access only the pieces of data necessary to answer the query.
- *Data Protection*: the use of encryption techniques, such as RSA or AES[12], to protect sensitive data at rest and in transit.

Privacy: *the right of an individual to decide what information about himself/herself should be communicated to others and under what circumstances* (Westin, 1970). Data leakage threats to privacy are caused by: disclosing data for purposes different from the one for which the data has been collected, revealing the PII intentionally or unintentionally, or by exposing sensitive information protected by data protection laws and regulations.

[12]Rivest, Shamir, and Adleman public key encryption technology (RSA) and Advanced Encryption Standard (AES) are widely employed encryption techniques.

Therefore, the privacy elements are as follows:

- *Purpose*: determines the reasons for the data to be collected or used. For example, the integration location should only grant the execution of a query if the purpose of the query specified by the data consumer matches the purposes for which the data has been collected.
- *Data sensitivity*: quantifies who should have access to data and how much harm would be done if the data is disclosed. For example, having security policies for data sources that determine the data's level of sensitivity. This helps in indicating which consumers are allowed to access that data. Therefore, data sensitivity can be employed if the data is classified.
- *Anonymity*: ensures that data does not reveal the identity of individuals. By restricting the exposure of PII or QIDs that lead to recognising identities.

Trust: *the belief that an entity will behave in a predictable manner by following a security policy* (Ross et al., 2014). Therefore, trust is used to quantify the risk of data leakage threats that materialise after the execution of queries is granted to data consumers, or the risk from an external entity handling the sensitive data. Once the data is disclosed to entities, they could misuse the data by sharing it with unauthorised parties, or use it for purposes other than allowed by the data providers.

Therefore, the main element under the trust property covered in this analysis is the ability to collect information about the trustworthiness of an entity and use that information in a trust model. The trust model helps in deciding whether the entity is allowed to access the data or not. In addition, the trust model can be used to assess the trustworthiness in data sources in providing reliable security policies.

4.5 Step 3: Identify Threats Violating Security Properties

The third step is to investigate the threats that violate the CPT properties discussed in Step 2. The threats are discussed by their location within the DIS architecture, covered earlier in Step 0.

1) Leakage from the the first layer - the data consumers' side: Threats occurring in this layer of the DIS, due to the violation of the CPT properties, shown as leakage 1 in Figure 4.3, are summarised as follows.

- Accidental authorisation to sensitive data due to the weakness, misconfiguration, or inappropriate choice of access control models (Braghin et al., 2003; Tipton and Nozaki, 2007; Watson, 2007; Pistoia et al., 2007), see Section 2.2.3.1.

- Secondary use of sensitive data. This leakage is caused by assuming trust in authorised data consumers and consequently not considering the risks associated with them in handling sensitive data (Paci et al., 2013), see Section 2.2.3.4.

- Using data for purposes different from the one for which the data has been collected, or for purposes against the ones indicated in the security and privacy policies (Guarda and Zannone, 2009; Clifton et al., 2004), see Section 2.2.3.1.

- Many types of intentional inference attacks, categorised as privacy attacks, can occur from proper and authorised consumers as well as external entities to infer sensitive data (see Section 2.2.3.1), the attacks are summarised as follows:

  1. Attribute-correlation attack: linking queries' attributes and predicates to infer sensitive information (Li et al., 2013).
  2. Consecutive queries to sensitive data sources (Bhowmick et al., 2006).
  3. Inference attack by record linkage: using non-confidential information and statistical aggregates to infer data (Clifton et al., 2004; Zhang et al., 2011).
  4. Inference attack by attribute linkage: linking QIDs together (Sweeney, 2002; Mohammed et al., 2011; Fung et al., 2012; Whang and Garcia-Molina, 2012).
  5. Inference attack by interval disclosure: computing missing values in sensitive data (Boyens et al., 2004).

2) Leakage from the middle layers: Threats occurring in this part of the DIS, shown as leakage 2 in Figure 4.3, are as follows.

- *Leaking sensitive data between the integration location layers.* This threat can occur either between the layers themselves, or between the layers and the platform that manages them (Herbert and Thieme, 2012).

- *Leaking sensitive data to external entities.* Leakage concerns are aggravated when any of the middle layers are provided by external entities, such as the use of cloud services or third parties to process the data where they cannot be fully trusted, see Section 2.2.3.2.

- *Leaking sensitive data by external entities due to flaws in security policies.* If the security policies of external entities that handle the data, such as backup databases of the DIS, do not exist, then there is a possibility of violating the confidentiality and privacy of the data (Meingast et al., 2006; Hashizume et al., 2013). In addition, the lack of governing transitive trust among the external entities may cause them to extend their rights to unauthorised parties, and hence expose the data (Fung et al., 2012).

- *Leaking sensitive data to the internet.* DIS uses the internet as a medium; therefore, there is a significant risk in leaking data to any entity on it, intentionally or unintentionally, by different types of security attack. These attacks are getting more creative, as hackers are now more skilled. One example of these attacks is exploiting a system to escalate privileges (Watson, 2007) and access confidential data.

- *Leakage caused by the integration approach.* The integration location includes the approach that achieves the combination of the data. Many approaches are presented in the literature, see Section 2.1.2. Some integration approaches consider privacy during the integration, such as privacy-preserving approaches, others have security and privacy outside their concern. The method by which the integration is conducted can impact data leakage. For example, the lack of dealing with the inapplicable confidentiality on merged data or the lack of legal knowledge on handling and merging the data (Batty et al., 2010) both cause leakage. Also, the lack of dealing with inconsistent data protection regulatory laws between countries and domains (Meingast et al., 2006) affects compliance, which is an important element of privacy, see Section 2.2.3.1.

- *Violating security policies of data sources from the lack of capturing those policies.* Each data source has its own security and privacy requirements and policies. Failing to capture those policies and to enforce them in the DIS may lead to violations of those policies and consequently cause data leakage. The integration approach and the integration location are both involved in capturing the data sources' policies and enforcing them on the integration process.

3) Leakage from the third layer - the data sources' side: The threats occurring from the data sources are as follows.

- *The lack of security and privacy information about the sensitive data causes leakage.* The DIS lacks instructions on how sensitive data should be processed and therefore violations to the confidentiality and privacy of the data can occur, i.e. failing to maintain the security and privacy of that sensitive data.

- *Inference attacks between data sources.* Data leakage between data providers arises from a special type of inference attack caused when data providers have pieces of the main data and they can infer the other pieces held by other providers (Goryczka et al., 2013). This leakage is shown in Figure 4.3 as leakage 3.

- *Inference attacks on data sources.* Using small samples of data sources recognised by deduction leaks data to unauthorised entities (van den Braak et al., 2012).

- *Lack of trust in data providers causes leakage.* Untrustworthy data sources may provide unreliable security policies that causes violations to security and privacy

policies of the data owners. The lack of trustworthiness criteria can hinder the process of indicating security policies violations. Trustworthiness needs to be assessed in external entities, such as cloud services, as discussed by Sun et al. (2011).

- *Lack of employing anonymisation techniques.* Data sources providing personal and sensitive data who do not consider applying any anonymisation techniques are at risk of data leakage by the DIS or its consumers if they do not handle the data properly.

4.6 Step 4: Analyse and Present the Findings

The final step of the analysis includes studying the threats found in the literature (Step 3), and using experts' responses alongside the confirmed DIS architecture. The threat analysis findings presented the following aspects:

- *Leakage Location*: the location of each threat (elicited in Step 3) within the DIS architecture (Step 0).
- *Leakage Source*: the entities responsible for the threat.
- *Vulnerability*: the weaknesses in the DIS that cause the threat.
- *Consequence*: the effect of the threats on the assets explained in Step 1.
- *Property*: which CPT property, from Step 2, is violated by the threat.

These findings are presented in Tables 4.4 to 4.8, where each threat is given a unique ID to simplify referencing.

4.7 Classification and Taxonomy of Data Leakage Threats

The data leakage threats resulting from the threat analysis discussed in Section 4.6 and detailed in Tables 4.4 to 4.8, are classified based on the security properties that guided the analysis (see Section 4.4). Figure 4.4 presents a taxonomy of the data leakage threats.

4.8 Data Leakage Threats and Databases Properties

Whether the type of the databases (i.e. data sources) integrated by a DIS is relational or non-relational, is not an area of concern in this study. The DIS is investigated in an abstract and high level manner that accommodates any type of database. In addition, it

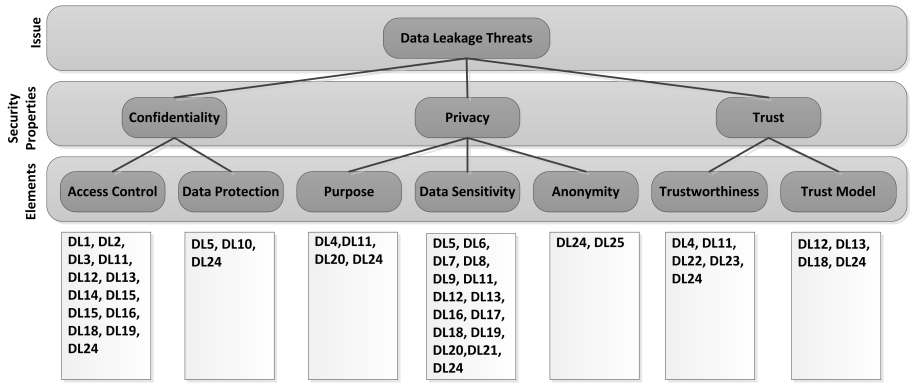

Figure 4.4: Classification and Taxonomy of Data Leakage Threats

is important to emphasise that the assumption of providing the databases to the DIS in the first place is for query purposes only. Hence, updating the databases is not expected to occur within the investigated DIS.

Several approaches to database properties exist to ensure data safety. One is the Atomicity, Consistency, Isolation, and Durability (ACID) goals of traditional database systems (Haerder and Reuter, 1983). ACID is usually used with SQL database and it guarantees data integrity and consistency immediately. The other approach is Basically, Available, Soft state, and Eventual consistency (BASE) approach that eventually guarantees the consistency of the data, and it is used with NoSQL databases (Sharma and Dave, 2012). Despite the differences in these approaches, the databases used within the investigated DIS do not create transactions that would add, delete, or update any of the records in the databases and the guarantee of data safety should be implemented by the data provider prior to providing the data for the integration.

To investigate the application and impact of ACID goals, for example, on the data leakage threats occurring in a DIS, it is essential to find a link between the goals and the threats. The link that we propose is through the implemented security properties. The threat analysis conducted in this chapter is guided by the Confidentiality, Privacy, and Trust (CPT) properties, as discussed Section 4.4. However, ACID goals are closely related to data integrity and consistency. This shows that the ACID goals and the data leakage threats do not seem to be directly related. However, non-ACID compliant databases may suffer the consequences of the lack of data durability in which the committed transactions do not remain in case of power loss. One would argue that this is a form of data leakage; however, it is not covered by the data leakage threats discussed in this chapter.

4.9 Summary

This chapter introduced the application of security by design through conducting an analysis to elicit data leakage threats in DIS. It started by creating a conceptualised DIS architecture, adapted from the literature, that was used as a guide to identify possible data leakage locations within the system. Expert reviews were conducted to discuss the leakage problem and confirm the appropriateness of the architecture and leakage locations. Following the findings of the experts, a refined version of the architecture with data leakage locations was proposed.

The confirmed architecture was used to determine the assets and the properties used, which consists of the CPT. This was followed by mapping data leakage threats to the architecture locations. The analysis resulted in 25 different data leakages threats occurring in the DIS, see Tables 4.4 to 4.8.

The threat analysis addressed research question RQ1, that aims to elicit data leakage threats and investigate them. The findings of this analysis are used for the mitigation of data leakage presented in the next chapter.

Table 4.4: Findings of the Threat Analysis conducted on the DIS Architecture

ID	Leakage Location	Leakage Source	Vulnerability	Consequence	Property	References
DL1	First Layer	Authorised/ unauthorised data consumers	Inappropriate choice of access control model.	Accidental authorisation to sensitive data.	C	(Braghin et al., 2003)
DL2	First Layer	Authorised/ unauthorised data consumers	Choosing a weak access control model.	Accidental authorisation to sensitive data.	C	(Tipton and Nozaki, 2007; Watson, 2007)
DL3	First Layer	Authorised/ unauthorised data consumers	Misconfigured access control.	Accidental authorisation to sensitive data.	C	(Pistoia et al., 2007)
DL4	First Layer and second layer	Authorised Data Consumers and external entities	Assuming trustworthiness of data consumers and external entities.	Secondary use of data by using it against its intended purposes or extending the rights to unauthorised entities.	PT	(Paci et al., 2013; Guarda and Zannone, 2009; Clifton et al., 2004)
DL5	First Layer	Any entity	The information about queries and predicates are in plain text.	Gathering and correlating query information to infer sensitive data about the query owner and the contents of the data sources (attribute - correlation attack).	P	(Li et al., 2013)
DL6	First Layer	Authorised data consumers	Allowing (uncontrolled) consecutive queries to sensitive data sources.	Infer sensitive data.	P	(Bhowmick et al., 2006)

Table 4.5: Findings of the Threat Analysis - continued

ID	Leakage Location	Leakage Source	Vulnerability	Consequence	Property	References
DL7	First Layer	Data consumers and other entities	The availability of non-confidential information and statistical aggregates relevant to the sensitive data.	Linking available data to infer sensitive data (inference attack by record linkage).	P	(Clifton et al., 2004; Zhang et al., 2011)
DL8	First Layer	Data consumers and other entities	Exposing all or most QID attributes.	Combining QID together or with external data to infer sensitive data (inference attack by attribute linkage).	P	(Sweeney, 2002; Mohammed et al., 2011; Fung et al., 2012; Whang and Garcia-Molina, 2012)
DL9	First Layer	Data consumers and other entities	Publishing part of the numerical sensitive data without an analysis of possible inferences.	Computing the missing values using already published data to infer sensitive data (inference attack by interval disclosure).	P	(Boyens et al., 2004)
DL10	Middle layer	Integration location	Lack of sensitive data encryption against the platform.	Leaking sensitive data or its combination to the integration location layers.	C	(Herbert and Thieme, 2012)

Table 4.6: Findings of the Threat Analysis - continued

ID	Leakage Location	Leakage Source	Vulnerability	Consequence	Property	References
DL11	Middle layer	Integration location - cloud services providers	Using public cloud services to process, store, and integrate sensitive data. These services are vulnerable to different threats by their nature.	Unauthorised access to personal and sensitive data.	CPT	(Carey et al., 2012; Ristenpart et al., 2009; Hashizume et al., 2013; Noor et al., 2013)
DL12	Middle layer	Integration location - third parties	Lack of security policy that clarifies the rights of trusted/ untrusted third parties on handling data, such as data backup.	Unauthorised access to personal and sensitive data.	CP	(Meingast et al., 2006; Hashizume et al., 2013)
DL13	Middle layer	Integration location - third parties	Lack of security policy governing transitive trust between external entities to handle data.	Unauthorised access to personal and sensitive data and possible trust violation.	CPT	(Fung et al., 2012)
DL14	Middle layer	Any entity	Access control issues (e.g. weakness, misconfiguration, etc.)	Escalate privileges to access sensitive data	C	(Watson, 2007)
DL15	Middle layer	Integration approach	Lack of dealing with inapplicable confidentiality on merged data.	Violates security policies by exposing data.	C	(Batty et al., 2010)
DL16	Middle layer	Integration approach	Lack of legal knowledge on data management.	Breaches of data protection regulations.	CP	(Batty et al., 2010)

Table 4.7: Findings of the Threat Analysis - continued

ID	Leakage Location	Leakage Source	Vulnerability	Consequence	Property	References
DL17	Middle layer	Integration approach	Lack of dealing with inconsistent data protection regulatory laws between countries or domains.	Violating law by exposing personal data.	P	(Meingast et al., 2006)
DL18	Middle layer	Integration Location and integration approach	Lack of capturing or applying the security and privacy policies of the data sources.	Lack of compliance to confidentiality and privacy requirements and hence violating the security policies.	CP	
DL19	Third layer	Data Providers	Lack of determining the needed security and privacy requirements.	Lack of capturing the requirements and causing information disclosure.	CP	
DL20	Third layer	Data providers	Allowing data providers to access the data of other data sources.	Accessing other pieces of data to derive sensitive information.	P	(Goryczka et al., 2013)
DL21	Third layer	Any entity	Using small and recognisable data sets	Deduction of sensitive and personal information.	P	(van den Braak et al., 2012)
DL22	All	Any entity including external entities	Lack of trustworthiness criteria (or the existence of a trust model) and lack of continuous evaluation of it.	Inability to assess the degree of security policy violation if trustworthiness is not assessed.	T	(Sun et al., 2011)

Table 4.8: Findings of the Threat Analysis - continued

ID	Leakage Location	Leakage Source	Vulnerability	Consequence	Property	References
DL23	Third layer		Lack of assessing the trustworthiness of a data source.	Incorrect data sources security policies.	T	(Sun et al., 2011)
DL24	All	Any entity	Lack of security management.	Lack of capturing security and privacy violations that may continue to occur in the system, and affect the trustworthiness of the entities involved.	CPT	(Sun et al., 2011)
DL25	Third Layer	Data providers	Lack of employing anonymisation techniques to protect PII.	Exposing the identity of individuals and possibly violating data protection regulations.	CP	

Chapter 5

A Framework for Secure Data Integration System

DIS are prone to several data leakage threats, as discussed in Chapter 4. This chapter addresses research question RQ2, see Section 3.1, that is concerned with proposing an approach for software engineers to mitigate data leakage. The chapter starts by presenting the Secure Data Integration Systems (SecureDIS) framework as an approach to secure DIS by design. The framework consists of six components: data and data sources, security policies, data consumers, the integration approach, the integration location, and System Security Management (SSM). Each component includes a set of guidelines designed specifically to mitigate the threats revealed by the threat analysis. The chapter presents the details of how SecureDIS was evaluated by experts in the field and how SecureDIS was extended and modified. Then follows a discussion on how SecureDIS is to be used by software engineers.

5.1 SecureDIS as an Approach Against Data Leakage

In order to mitigate data leakage threats in DIS, the design of the DIS from the start needs to be secure by following the security by design concept (McGraw, 2004). The concept suggests propagating security requirements from the early stages of development (Mouratidis et al., 2005) through to the end. One of the approaches to develop secure software that mitigates risks is by understanding the threats and attacks and cycling them back to development, which was outlined in Chapter 4. In addition, this research conjectures that in order to mitigate the data leakage threats identified, a combination of Confidentiality, Privacy, and Trust (CPT) should be considered when designing a DIS.

Since the threats have now been identified, the aim is to assist software engineers to build a secure DIS that mitigates those threats. The mitigation approach aims to address the main architectural components of a DIS with middle layers in the form of a framework named SecureDIS. The framework includes guidelines to secure the DIS by design.

Ideally, the integration process should occur at two different levels: the policies level and the data level. The policies level addresses the security and privacy requirements that belong to each data source. The data level is concerned with the actual integration. The integration process should therefore ensure the data level is influenced by the policies level to guarantee the data is exposed according to its policies.

A framework approach to create SecureDIS was adopted for two reasons. The first is that previous approaches were generally presented as frameworks, such as the works of Bhowmick et al. (2006) and Clifton et al. (2004). This allows the approaches to be flexible during application. The second reason arose from discussions with experts, see Section 4.2.4. They recommended proposing a framework to mitigate data leakage that can encompass different aspects of the DIS.

The reason that SecureDIS contains guidelines is to harmonise it with the studies, reviewed in Section 2.4.3, that aimed to create and construct security guidelines. Hence, the security guidelines generated within SecureDIS aim to achieve a similar goal.

5.2 The Preliminary SecureDIS Framework and Guidelines

DIS are usually complex (Russom, 2008), and to simplify the mitigation approach for those systems, the scope of SecureDIS should be defined, see Section 1.2. The framework and its guidelines target software engineers who analyse and design DIS and plan to secure it from the early stages of development.

To build the SecureDIS framework, the works of Clifton et al. (2004), Bhowmick et al. (2006) and Haddad et al. (2012) were carefully investigated for the following:

- to understand the DIS architecture and its components. That resulted in the components: data and data sources, the integration location, the integration approach, and data consumers. In Chapter 4, two additional components were added: security policies and System Security Management (SSM), see Section 4.2.7.
- to contribute to the understanding of the data leakage threats and the ways of mitigation that are possible in DIS. This was combined with the threat analysis findings, see Tables 4.4 to 4.8, to formulate the SecureDIS guidelines.

The SecureDIS guidelines were created by combining those found in the literature with an understanding of the properties of CPT. The guidelines provide a list of *"what"* should be done to mitigate data leakage rather than *"how"* it is implemented. Each guideline is mapped to multiple data leakage threats, see Tables 4.4 to 4.8. The mapping was based on each threat's source and location and its relevance to the SecureDIS components. Also, each guideline indicates its relationship with the CPT properties, see Section 4.4. To include a guideline under a specific component, each guideline was inspected in relation to other guidelines in different components. This process of refining the guidelines was iterative, as many guidelines were moved around components and grouped differently until the final version was reached.

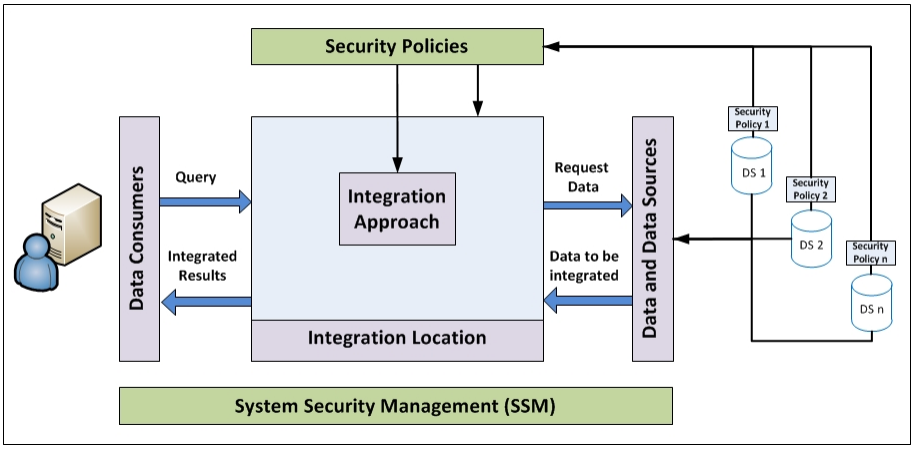

Figure 5.1: The SecureDIS Components

Figure 5.1 shows SecureDIS architectural components and their interaction, each component is explained as follows:

The following discuss each component. The preliminary guidelines can be found in

1- Data and Data Sources Data sources contain the actual data used in the integration process to resolve consumers queries, see Section 2.1.1. Data sources are important from a security perspective as discussed in Section 4.2.2.

Data sources security depend on the provision of sufficient security and privacy requirement data (or meta-data) to ensure that the data is disclosed in a secure fashion. The security meta-data is relevant to the data-centric security criteria, described in Section 2.3.1, and includes:

- Data owner who deletes, modifies, and enforces regulations.
- Data location, country and organisation.
- Internal and external entities allowed to access the data.
- Access control used.
- Encryption requirements for data in storage and in transit.

Data sources privacy requirements, see Section 2.2.1, include:

- Data destruction/retention period.
- The method of sharing personal data.
- The allowed purpose of using the data (purpose statement).
- The level of data allowed to be used, e.g. exact value, aggregate, range, attributes, etc.
- The level of data to be shared for each purpose.
- The allowed level of legitimate information exposure with privacy loss.

For the identity attributes, such as PII or QIDs, anonymisation techniques can be used depending on the integration context, see Section 2.3.2. Some sensitive data sources need to be removed completely from a DIS, as they may represent a recognisable sample that has a high potential of being inferred (van den Braak et al., 2012).

To make use of the security and privacy requirements from the data sources, security and privacy policies for each data source need to be created. This aims to maintain the requirements of those data sources and ensures the integration process continues to comply with these policies.

Information about data sources' trustworthiness needs to be collected to ensure they are suitable for participation in a secure DIS and to assess the reliability of their security policies. This is to overcome data leakages DL22 & DL23 (see Tables 4.7 and 4.8). The proposed indicators of trustworthiness are as follows.

- Certification by trusted entities or organisations that check the credibility of data sources.
- Data freshness, depending on its regular patterns of updates.
- Frequency of data source use by other entities.

Table B.1 in Appendix B shows the guidelines for this component.

2- Security Policies In the context of data integration, security policies were discussed in Section 2.4.4. In SecureDIS, a security policies component is proposed to define a policy for the whole system, which encompasses the security and privacy policies of the DIS and of the data sources, in addition to the data protection regulations that a DIS is required to enforce.

The security policies are used to guide the integration approach to ensure the security and privacy requirements of the entities involved are maintained. Consumers' queries are resolved in line with those policies, and therefore this is an essential part of any DIS.

In addition, this component includes the trust models used in the DIS, and governs data disclosure to other entities, such as data providers (Goryczka et al., 2013), third parties, and cloud services (Fung et al., 2012; Meingast et al., 2006). The policies aim to protect against unauthorised access and limit sensitive data disclosure, as discussed in Section 4.5. Table B.2 shows the guidelines for this component.

3-Data Consumers This component represents the location of the requests for access to the integrated data using queries, see Section 2.1.1. Several data leakage threats occur in this location, as discussed in Section 4.5. One of the approaches to counter these threats is to conduct a query analysis within the integration location to check the predicates and types of data returned to identify possible security and privacy breaches before they occur (Bhowmick et al., 2006). In addition, the conformity with access control rules, and security and privacy policies, applies intensively to this part of the system. Table B.3 in Appendix B shows the guidelines for this component.

4- The Integration Approach This component represent the chosen method to combine the data from multiple data sources, as explained in Section 2.1.2. Approaches differ in their consideration of privacy and security. Some approaches may cause data leakage, as explained in Section 4.5, and some preserve a certain property, such as privacy.

Before the integration process begins, implementers should consider the security policies defined for the DIS and perform the integration in light of these policies (Haddad et al., 2012). During integration, several security and privacy steps can be undertaken to ensure data leakage is prevented, such as analysing the query results to predict possible security and privacy violations and eliminating them before returning the results to the consumer (Bhowmick et al., 2006). Also suggested is annotating the data resulting from queries with proper security and privacy meta-data. This helps data consumers be aware of the expected level of security and privacy and helps them to understand the data value and the consequences of its misuse. Table B.4 in Appendix B shows the guidelines for this component.

5- The Integration Location The integration process takes place in this component, see Section 2.1.3. The integration location is susceptible to data leakage, as discussed in Section 4.5, as it may consist of multiple layers and may involve external entities. To mitigate these threats, the integration location is required to comply with the security and privacy policies of the system (van den Braak et al., 2012), as well as the data sources policies, to protect the confidentiality and privacy of the data before and after the integration is achieved (Youssef and Alageel, 2012). Table B.5 in Appendix B shows the guidelines for this component.

6-System Security Management (SSM) Failing to manage DIS by monitoring and responding to security-related activities that may affect the security and privacy of the data leads to data leakage, see Section 2.4.1. As discussed in the proposed architecture in Section 4.2.7, the SSM is introduced to SecureDIS for several reasons: 1) it contains

security activities to manage and ensure achieving the DIS security goals by monitoring and logging data access; 2) it conducts audits to ensure compliance with regulations, licences, and policies (Hennessy et al., 2009; Youssef and Alageel, 2012); and 3) SSM is responsible for configuring the access controls properly to include privacy and security requirements (Takabi et al., 2010; Hennessy et al., 2009). Table B.6 in Appendix B shows the guidelines for this component.

5.3 Expert Reviews Design

The preliminary SecureDIS guidelines require a method of confirmation. Expert reviews were one of the chosen research methods for the confirmation as they can combine quantitative and qualitative approaches (see Section 3.2.2). The following subsections detail this process.

5.3.1 Purpose

The main purpose of these reviews is to validate the preliminary SecureDIS framework and guidelines presented in Chapter 5. The quantitative and qualitative analysis aim to address research question RQ2 concerned with assisting software engineers in building secure DIS by design, see Section 3.1. Table 5.1 lists a number of purposes that help in achieving RQ2, each of which is given an identity for use within this chapter. Each purpose shown will later be discussed with relevance to the analysis of the quantitative and qualitative results of this review.

5.3.2 Material Presented

The material presented to each expert contains the following (Appendix C.2 has the details).

- A description of the SecureDIS framework, see Section 5.2, and the CPT properties as explained in Section 4.4.
- An architectural diagram of the SecureDIS framework, as presented in Figure 4.3, illustrating data leakage locations.
- A detailed table of data leakage threats, as presented in Tables 4.4 to 4.8 .
- A questionnaire to validate the proposed guidelines under each component, discussed in detail in Section 5.3.2.1.

Table 5.1: Detailed Purposes of the Second Expert Reviews

Purpose	ID
To include data leakage threats that were not covered by SecureDIS	P1
To improve SecureDIS by discussing its components and the CPT properties	P2
To confirm/reject the SecureDIS guidelines	P3
To expand the proposed guidelines by obtaining more details from experts	P4
To discuss other mitigation guidelines that were not covered by SecureDIS	P5
To acquire knowledge about current practices in the field of data security. This leads to understanding the value of SecureDIS in terms of what is currently available, in addition to finding out whether it is comprehensive and practical.	P6
To gather experts' ideas for further improvements	P7

- A questionnaire to evaluate the overall coverage of CPT by SecureDIS, discussed in detail in Section 5.3.2.1.
- Open-ended questions used in the interview, discussed in detail in Section 5.3.2.2.

The following sections discuss the design of the methods used for data collection.

5.3.2.1 Questionnaires

Questionnaires are the quantitative part of this activity. They are useful to get a common and measured perspective among all the experts. In this review, questionnaires were used to validate the guidelines and to evaluate SecureDIS in general. Questionnaires are used here to mainly serve purposes P2 and P3 presented in Table 5.1.

To validate the guidelines, for each component, the corresponding set of guidelines were presented to the experts; each guideline was validated individually. The purpose of this questionnaire is to accept or reject the proposed guidelines. To ensure the validity of responses and to give experts a choice of selection, a five-point Likert scale was used (Strongly Agree, Agree, Neither Agree nor Disagree, Disagree, Strongly Disagree) (Likert, 1932).

Table 5.2 below describes the mean intervals in relation to the Likert scale that will help in accepting or rejecting the experts' responses. The length of the interval was

determined by calculating: 5 (the last Likert point) - 1 (the first Likert point) = 4, then the number was divided by the number of scales, $4/5 = 0.80$.

Table 5.2: Mean Intervals for the Likert Scale

Mean Interval	Likert Scale
1.00 - 1.80	Strongly Disagree
1.81 - 2.60	Disagree
2.61 - 3.40	Neither Agree nor Disagree
3.41 - 4.20	Agree
4.21 - 5.00	Strongly Agree

According to the means presented in Table 5.2, for each guideline presented to the expert, if the overall experts' responses were ≥ 3.41, the guideline was accepted, otherwise the guideline was rejected. If a guideline was questioned during validation, it created an opportunity for further discussion and improvement. If the rating of the guideline showed disagreement, the expert was asked for reasons as well as suggestions for improvement.

To evaluate SecureDIS in general, four questions were asked about SecureDIS with the same Likert scale discussed above. The questions were:

1. Do you think that SecureDIS covers data confidentiality?
2. Do you think that SecureDIS covers data privacy?
3. Do you think that SecureDIS covers trust within its components?
4. Do you think that the SecureDIS components are suitable?

The aim of these questions was to confirm that SecureDIS actually covers the CPT properties to mitigate data leakage and that the proposed components are suitable. In case of disagreement, the expert was asked for more detail as to how to improve the framework.

5.3.2.2 Open-ended Questions

Open-ended questions are the qualitative part of this activity. Experts were asked several questions that were derived from specific purposes. The experts' feedback was recorded and analysed according to themes. Table 5.3 shows the details of each question asked and its intended purpose.

5.3.3 Ethical Approval

Since the selected research methods require people as participants, ethical approval to conduct this study was obtained from the university. It extends the previous research

Table 5.3: Open-ended Questions Linked to Purposes of the Second Expert Review

Open-ended Questions	**Purpose**
Q1: What sort of security guidelines, standards and best practices related to data security are used currently in your organisation?	P6
Q2: What do you think the reason for this choice of guidelines that you use or that others use in the domain of data security?	P6
Q3: Looking at the data leakage threats covered by SecureDIS, do you have any more threats to include? If yes, please elaborate	P1
Q4: Are you aware of any data leakage threats in your organisation? [Please note it's a yes or no question]	P6
Q5: If you answered Q4 with yes, have you implemented any countermeasures against those threats?	P5 P6
Q6: From your experience, which of the following can be used as criteria of trustworthiness for both data and data sources? [You can select more than one] - Update-frequency of a data source (data freshness), - The frequency of data source use by other entities, - Certification by some trusted entity, - Other, specify	P5 P4
Q7: Reading the guidelines, do you find them similar to other standards, guidelines, and best practices that you know?	P6
Q8: Are there any more aspects of DIS concerning CPT that were not considered by SecureDIS?	P1, P2, P5
Q9.a-Q9.c: Seeks expert's opinion on whether SecureDIS has covered confidentiality, privacy, and trust.	P7
Q9.d: Investigates the suitability of SecureDIS components.	P2
A final question asking for further improvements	P7

ethics number ERGO/FPSE/8911. No personal data was collected during this study. However, information collected was anonymised and any corporate identifying information was removed.

5.3.4 Recruiting Participants

To obtain significant results from the quantitative data, it was necessary to calculate the number of experts needed for this research activity. The G*Power[1] tool was used for the calculations, using an effect size = 0.8, Alpha = 0.05, and Power = 0.87. G*Power predicted the number of experts needed was 14.

[1]GPower is an open source program for power analysis and sample size calculations, and can be obtained from http://www.gpower.hhu.de/en.html

Selection criteria were established for recruiting experts for participation. Experts needed to be from one or more of the following areas:

1. Software engineering: experts were system analysts, developers, project managers, or researchers.
2. Security and privacy: experts were consultants, practitioners, or researchers.
3. Data management and databases related fields: experts were database administrators or researchers.

Such diverse areas of expertise aimed to provide more practical feedback in considering the important aspects of DIS and also to remove bias towards a specific field. Based on the maximum years of experience, each expert was identified with a domain and given an ID to be used throughout this chapter. Table 5.4 shows their qualifications.

Table 5.4: Experts' Areas of Expertise

No	**Expert ID**	**Area of Expertise**	**Method of Interview**	**Domain**
1	Expert A	Senior security and privacy consultant	Internet	Industry
2	Expert B	Senior software engineer, with experience as a system analyst and a developer	Internet	Industry
3	Expert C	A practitioner and researcher in security and also a software engineering researcher	In person	Academia
4	Expert D	Privacy researcher	In person	Academia
5	Expert E	Software engineering researcher, with experience in systems analysis	In person	Academia
6	Expert F	Security and law researcher	In person	Academia
7	Expert G	Security researcher	In person	Academia
8	Expert H	Security, researcher and developer	In person	Industry and academia
9	Expert I	A researcher in data analysis and visualisations	In person	Industry and academia
10	Expert J	Software developer	In person	Industry
11	Expert K	Data management and data integration technical consultant	Internet	Industry
12	Expert L	Database administrator	Internet	Industry
13	Expert M	Systems developer and researcher in open data and semantic web	In person	Industry and academia
14	Expert N	Senior cloud-security consultant	In person	Industry

5.3.5 Piloting Expert Reviews

A pilot of the proposed material was undertaken using three researchers in software engineering and security. Improvements were made to diagrams, illustrations, and the wording of the guidelines. More instructions were added to clarify the required tasks. The material was ready to be presented to experts by the end of pilot stage.

5.3.6 Review Procedure

According to the criteria mentioned in Section 5.3.4, the experts were contacted either personally or by email and provided with an information sheet to describe the purpose of this study. Appendix C.1 shows a sample of the email message sent to the experts, along with the information sheet and the consent form. Appointments were made spanning the period between June 2014 and August 2014. The time frame allocated for each expert review was between 45 to 60 minutes. Experts were interviewed in person or via the Internet.

Each expert was presented with a consent form to sign, and then given a prepared document containing a brief explanation of the SecureDIS framework and the material discussed in Section 5.3.2.

After a brief discussion with each expert of their area of expertise and number of years of experience, a thorough explanation was provided of the components of SecureDIS and their position in the architecture. Data leakage threats were described using the diagram that highlights locations of leakages. The explanation took from 10 to 15 minutes to complete.

Following the explanation, a questionnaire about each component of the SecureDIS guidelines was presented. Experts spent 5 to 10 minutes on each component. After reading the guidelines and filling in the questionnaire, questions about the guidelines were asked and disagreements about the wording, the order, or the content of each guideline were thoroughly discussed. At the end of the review, several open-ended questions were asked and comments were recorded for the purpose of further improvement.

5.3.7 Collecting and Analysing Data

Most of the data collected from the experts' discussions were qualitative. The reviews were recorded, transcribed, and then deleted to protect individual and corporate privacy. A sample of a transcribed interview is given in Appendix C.3. Other data collected from the questionnaires were quantitative; hence, a statistical tool was used to analyse the data. Both types of data analysis are described below.

5.3.7.1 Qualitative Analysis

To qualitatively analyse experts' responses to open-ended questions and their comments on the guidelines, the interviews were transcribed and saved into NVivo[2]. NVivo is a software tool used to manage and understand textual data and make the most of it. Experts' responses were tagged using NVivo, according to analysis themes and guidelines. The responses were categorised into themes, see Section 3.2.2.

5.3.7.2 Quantitative Analysis

To statistically analyse quantitative data, experts' responses were collected and entered into SPSS[3]. The One Sample t-test was used to analyse the results of the quantitative data. One Sample t-test helps in comparing the mean of a population (μ) with a hypothesised value ($\mu 0$). The hypothesised mean is ($\mu 0$) = 3.41, according to the discussion in Section 5.3.2.1. This assists in rejecting or accepting a guideline.

The null hypothesis used in this test is $H0 : \mu < \mu 0$, while the hypothesis is $H1 : \mu >= \mu 0$. The statistical significance level alpha (α) was set to 0.05: if the statistical significance for each guideline is greater than (α), the null hypothesis is accepted and the guideline is rejected; otherwise the guideline is accepted.

5.3.8 Expert Reviews: Benefits and Limitations

The results of these reviews were useful as they have given an opportunity to discuss SecureDIS and obtain practical feedback on methods of improvement. In addition, the reviews indicated several factors influencing the applicability of the guidelines in practice. Experts confirmed the validity of the guidelines which was useful in answering the research questions. The changes suggested were incorporated in the improved version of the framework.

Unfortunately, the reviews raised the following challenges:

- The difficulty of finding experts willing to take part in such interviews.
- The difficulty in getting appointments with experts who are usually busy.
- Some experts were reticent about data leakages and improvements, as they perhaps thought it might violate their corporate privacy.
- The insights of the experts working in industry who have a more product-oriented perspective toward the architecture of SecureDIS.

[2]NVivo can be obtained from: http://www.qsrinternational.com/product
[3]SPSS can be obtained from: http://www-01.ibm.com/software/uk/analytics/spss/index.html

To overcome these limitations, several actions were taken. First, academics in the relevant area of research were contacted to assist in finding qualified experts who were willing to take part. Secondly, experts were given the option for an online interview at a time convenient to them. Thirdly, experts were reassured that the questions asked during the interview were relevant to the architecture given. However, questions relevant to data leakage threats occurring currently in their organisations were directed towards mitigating threats rather than exposing the organisation's security weaknesses. Finally, experts from the industry were reminded frequently during the interview that SecureDIS is conceptual and it represents a high level understanding of DIS. This helped in directing them to think about the general theory behind the system.

5.4 Results and Findings

This section explains the confirmation of the SecureDIS framework and guidelines. In addition, it discusses the qualitative analysis of the experts' responses to open-ended questions. The analysis themes are discussed in the following subsections.

5.4.1 Data Leakage Threats

This theme embraces the qualitative results for questions Q3 and Q8 of the reviews that serve purpose P1. Most experts thought that the data leakage threats covered were sufficient. However, some experts identified several new threats to be included. Expert B discussed a scenario in which data leakage could occur between data consumers. Data consumers who are able to see which data sources are queried and which meta-data is targeted are prone to carry out inference attacks. Expert L also identified data leakage threats from the consumer side. Expert C claimed that the easiest way to attack a system is by attacking the consumer. This can occur, according to expert I, by having cumulative information stored on users' computers. This makes it necessary to provide user-side security as part of the prevention mechanism.

Expert A pointed out an interesting type of data leakage, where vendors providing the means of integration, either hardware or software, are accessing sensitive data. The expert says: *"there are major data leakage threats from our IT technology providers (hardware, and software), such as back-doors, patches, support, e.g. Microsoft patches, Oracle loop-holes and back-doors. This is a major risk and it needs to be addressed as part of an overall Risk Management Programme, under the section Vendor Management. Remember, IT suppliers have connections to their government security agencies"*.

In addition, experts B and F highlighted the threat of insiders. Expert B was of the opinion that most employees are to be trusted; however, untrusted ones are able to bypass an organisation's boundaries and misuse the data. Expert F pointed out that a

malicious insider can use the sensitive data or sell it for personal interests. This could happen intentionally, or unintentionally through social engineering (expert M).

Expert N provided the following additional threats: broken trust models and trust violation threats, failing to profile human activities, and the lack of security policy maintenance, e.g. how often does entity x checks y? Or on every element y, we need to perform this method z.

On *risk assessment*, expert J suggested the use of these assessments to identify possible risks. Expert I mentioned the need to define the risks associated with the trust models in guidelines 25 and 26. Expert A suggested considering periodic risk assessment of vendor products to *"ensure that we do not have an invisible elephant within our IT operations or infrastructure components"*.

On *secure transmission*, expert A discussed leakage threats occurring at IP address level. The expert suggested monitoring and logging the handover of information to the consumer by having carrier precautions that include agreed methods, such as encryption agreements. Expert H suggested considering secure connections and secure protocols, such as HTTPS. Expert I's company uses only company products that are designed to lock down connections to the cloud to prevent leakages to unauthorised entities, which ensures that no data transmission occurs. Expert J used several techniques at the network level, one for SMTP/HTTP blocking by integrating with proxy servers, and another through Message Transfer Agents (MTA), as an intermediate countermeasure against data leakage. The expert mentioned that the status of the data defines the proper countermeasures, for example: data in motion over a network or Internet, data in use at endpoints of a network, and data residing in the file system, databases or other storage systems.

5.4.2 Summary of SecureDIS Guidelines Reviews

This theme serves purposes P3, P4 and P5. Experts discussed the guidelines in detail. Combining the results of the quantitative and qualitative analysis, the guidelines were improved in the following respects: acceptance or rejection of a guideline, rewording a guideline, adding details and conditions to a guideline, and proposing a new guideline (evident in responses to Q5, Q6 and Q8 of the review).

The following sections summarise the changes made to each component of SecureDIS. The numbering of the newly included guidelines are based on the name of the component. The confirmed set of guidelines resulting from the completion of the expert reviews, and their comments, as well as details relating to each guideline are addressed later in this chapter, see Section 5.6.

5.4.2.1 Data and Data Sources Component

Before discussing the guidelines of this component, the trustworthiness of data sources was discussed thoroughly with experts as a response to question Q6. This was to provide criteria to indicate trustworthiness of data sources, which serves purposes P4 and P5. Three trustworthiness indicators were suggested to experts as initial criteria. The proposed indicators were accepted by the experts, but they thought that the indicators did not necessarily apply to all data sources. There was a need to consider the entity providing the data, whether it was a known or unknown organisation (experts C and E). Complete data provenance was required, according to experts M and N. In addition, the purpose of providing the data should be present (experts H and E). There was a need to track the reputation of the organisation providing the data, as expert H suggested. For example, Facebook's record in collecting personal data can be taken as a sign of untrustworthiness.

On the *update-frequency* indicator, expert G pointed out that the update frequency of some types of data source are not necessarily an indicator of their trustworthiness. For example, data sources regarding scandals are frequently updated but are not trustworthy. Likewise, expert I mentioned that a historical data source was not always fresh and up-to-date and it was still counted as trustworthy. On the other hand, experts B and K thought that, depending on the nature of the data source and its freshness pattern, this indicator could be helpful, especially when the date of the last update was provided with the dataset, as expert H suggested.

Regarding *the frequency of use by other entities* indicator, expert F believed that if well-known entities were using that data source, then the DIS could use it as well. However, experts K and H said that frequency of use is not enough of an indicator, as a data source might be one of a kind and not frequently used, yet trustworthy. Expert J disagreed that frequency of use is a sign of credibility and therefore it was a useful indicator. Expert G believed credibility required an in-depth analysis using other resources.

Expert H commented on the certification indicator with *"do I actually trust anybody?"* and saw certification as problematic. However, expert A held the opinion that certification by an entity is needed, because *"it will ensure that due diligence has been made prior to the issuance of this certificate. Also, it will be protecting the firm from any liability in the future, if any legal case arises. No data should be accepted without certification or a waiver for not being certified"*. Experts L and E agreed with the importance of that indicator. However, expert F did not think that an entity should be truly trusted to certify a data source; otherwise, this indicator was not sufficient.

Several experts suggested adding *technical indicators* to determine trustworthiness, including:

- Location of the data: where is it hosted, a government website or a personal website? (expert M).
- Data is of good quality, maintained and valid (experts M & N).
- Data is available (expert J).
- Data is well kept and backed-up frequently (experts C & I).
- Data is provided through secure APIs and secure channels (experts K, M, and I).
- The security of the eco-system around the data should be evaluated (expert N).
- The use of cryptography and hashing functions to protect the data (expert F).
- Licences on which data is to be shared (applicable in open data, for example) (expert I).
- Providing more information of what is required from the data consumer to keep data privacy can be another indicator (experts D & N).

Another suggested indicator of trustworthiness was the compliance of the data provider with well-known standards (experts C and N), or with the data privacy requirements and regulations of the country from which the data was coming (experts D and I). Expert B raised the question *"when is trustworthiness checked? Is it at query time?"*. Expert D suggests checking trustworthiness periodically to be safe. Trustworthiness checks should be made automatically (expert B), at a specific time.

One interesting thought from expert B was that it is probably better to search for untrustworthiness rather than for trustworthiness. Expert B commented, *"sometimes we need to ask for evidence of untrustworthiness!"* especially when a data source is a correlation of other data sources.

The results and findings of the reviews on the guidelines of this component are as follows.

a) The quantitative results The results of applying one sample t-test to the guidelines 1 through 6 are shown in Tables 5.5 and 5.6. The results show that the statistical significance $p < 0.05$; therefore, $H0$ is rejected. This means that experts accept the guidelines of this component.

b) The qualitative findings Modifications were made to the guidelines, as discussed below.

Guideline 1: Was modified to focus on security meta-data only. One element was added to the security meta-data fields: *Data access location (inside/outside an organisation or a country)* (Expert M).

Table 5.5: One-Sample t-test for Data and Data Sources Guidelines

Guideline	N	Mean	Std. Deviation	Std. Error Mean
1	14	4.6429	.49725	.13289
2	14	4.5714	.64621	.17271
3	14	4.5714	.64621	.17271
4	14	4.5000	.51887	.13868
5	14	4.4286	.85163	.22761
6	14	4.7857	.42582	.11380

Table 5.6: One-Sample t-test for Data and Data Sources Guidelines

	Test Value = 3.41					
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Guideline 1	9.277	13	< .001	1.23286	.9458	1.5200
Guideline 2	6.725	13	< .001	1.16143	.7883	1.5345
Guideline 3	6.725	13	<.001	1.16143	.7883	1.5345
Guideline 4	7.860	13	< .001	1.09000	.7904	1.3896
Guideline 5	4.475	13	.001	1.01857	.5269	1.5103
Guideline 6	12.088	13	<.001	1.37571	1.1299	1.6216

Guideline 2: Was modified to focus on privacy requirements in general, including the data use purpose statement. One element was added to the requirements: *Sensitivity and privacy level (such as intellectual property, trade restriction, embargoed information or privacy properties)* (Expert C).

Guideline 4: Expert I raised a practicality issue regarding this guideline; hence the following technique was added to the details to overcome that: *Create a data map to indicate which attributes may be used to link other attributes in other databases to ensure changing/removing them.*

c) Added guidelines Two guidelines were added to this component, based on the discussion with experts.

Guideline DS1: *Determine the trustworthiness of data sources by monitoring different aspects (e.g. entity type, update frequency, technical properties).*

Guideline DS2: Expert B, *Provide feedback to data providers about the use of personal data, which may include:*

- *The purpose of using data,*
- *the entity that used the data (or the role name),*
- *the part of the data that was used,*
- *and the date and time of use.*

5.4.2.2 Security Policies Component

The security policies component was one that generated considerable discussion with some experts. The proposed guidelines of this component were discussed in Table B.2.

a) The quantitative results The quantitative results of applying one sample t-test to guidelines 7 to 12 are shown in Tables 5.7 and 5.8. The results show that the statistical significance $p < 0.05$; therefore, $H0$ is rejected. This means that experts think the guidelines for this component are acceptable.

Table 5.7: One-Sample t-test for Security Policies Guidelines

Guideline	N	Mean	Std. Deviation	Std. Error Mean
7	14	4.7857	.42582	.11380
8	14	4.5714	.85163	.22761
9	14	4.7857	.57893	.15473
10	14	4.3571	.92878	.24823
11	14	4.1429	.86444	.23103
12	14	4.3571	.92878	.24823

Table 5.8: One-Sample t-Test for Security Policies Guidelines

	Test Value = 3.41					
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Guideline 7	12.088	13	<.001	1.37571	1.1299	1.6216
Guideline 8	5.103	13	<.001	1.16143	.6697	1.6531
Guideline 9	8.891	13	<.001	1.37571	1.0414	1.7100
Guideline 10	3.816	13	.002	.94714	.4109	1.4834
Guideline 11	3.172	13	.007	.73286	.2337	1.2320
Guideline 12	3.816	13	.002	.94714	.4109	1.4834

b) The qualitative findings Expert L raised concerns over defining security responsibilities of the organisation, and the obligations to protect data after it has been fully received. In addition, expert L suggested defining clearly that any data loss that occurs before the data is fully received is not part of the agreement.

The guidelines for this component were changed as follows.

Guideline 7: Extended by expert B to read: *"Ensure that the security policy considers the integrated security policies of the participating data sources and the DIS itself, including the platform"*. Details of this guideline were added as follows:

- *Consider the trade-off of correlation vs. the quality of data, and how lowest policies affect the highest* (experts B & I).

- *Consider how to enforce the security policy* (expert F).
- *Ensure that the security policy covers the conflicts between data privacy law in one location and other locations, such as USA and Europe* (expert B).

Guideline 8: Reworded by expert N to read: *"Define third parties' and cloud services providers' (public, private, hybrid, and community) rights on data (including transitive trust)".*

Guideline 10: Expert M thought the guideline required further clarification. Other experts provided details for this guideline:

- *Define trust by level* (expert F).
- *Trust model can be qualitative by nominating trustworthy people/entities to handle critical data* (expert K).
- *Trust models can depend on 1) time factors and 2) event factors* (expert I).
- *Define how trust is enforced* (expert F).
- *Continuously re-evaluate trust models and monitor changes* (experts I and D).

Guideline 11: Experts suggested the following details.

- *Consider cost vs. value of implementing this guideline* (expert B).
- *Add a flag for the SSM component, in case of violation* (expert H).

Guideline 12: Experts J and N modified this guideline to read: *"Define data consumer's consumption rights, considering trust as one parameter that changes data access".*

c) Added guidelines: The following are new guidelines added to this component.

Guideline SP1: *Define how compliance with policy or lack of it is enforced* (expert N).

Guideline SP2: *Test the proposed security policies before implementation* (expert B).

Guideline SP3: *Consider the risks associated with the violations of security policies and the trust model* (expert I).

Guideline 31: This guideline was moved from the SSM component to this component as it seems more suitable according to expert H. The guideline was reworded by expert N to read: *Where using cloud services to handle data (i.e. access, process, store or manage data), establish the required trust to achieve those tasks.*

5.4.2.3 Data Consumers Component

a) The quantitative results The results of applying one sample t-test to guidelines 13 to 17 are shown in Tables 5.9 and 5.10. The results show that the statistical significance for guidelines 13, 14, 15 and 17, $p < 0.05$; therefore, $H0$ is rejected. However, the statistical significance for guideline 16, $p > 0.05$; therefore $H1$ is rejected. This means that experts accepted all the guidelines of this component except guideline 16, which should be excluded.

Table 5.9: One-Sample t-test for the Data Consumers Guidelines

Guideline	N	Mean	Std. Deviation	Std.Error Mean
13	14	4.5714	.85163	.22761
14	14	4.5714	.51355	.13725
15	14	4.4286	.85163	.22761
16	14	4.0714	1.32806	.35494
17	14	4.5714	.64621	.17271

Table 5.10: One-Sample t-test for the Data Consumers Guidelines

	Test Value = 3.41					
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Guideline 13	5.103	13	<.001	1.16143	.6697	1.6531
Guideline 14	8.462	13	<.001	1.16143	.8649	1.4579
Guideline 15	4.475	13	.001	1.01857	.5269	1.5103
Guideline 16	1.864	13	**.085**	.66143	-.1054	1.4282
Guideline 17	6.725	13	<.001	1.16143	.7883	1.5345

b) The qualitative findings Experts suggested the following.

Guideline 13: Reworded by expert N to read: *"Attempt resolving queries only when after access to data is granted"*.

Guideline 14: Extended by expert C to read: *"Analyse the features of the query, e.g. type of predicates, types of data returned, to identify possible security and privacy breaches and to conform with the security policy"*. Expert I discussed a practicality issue with this guideline. Hence the following detail was added: *The analysis is compared with previously studied data leakage threats resulting from threat analysis.*

Guideline 15: Reworded by expert N to read, *"Keep a record of all queries and classify them according to 'type of threat' to prevent consecutive queries inference attacks"*. Details of this guidelines are as follows:

- *The 'type of threat' should be determined by best practices in the field as it is a complex process* (expert I).

- *Determine the appropriate response to this action* (expert C).
- *The analysis should be a short-time behaviour one to become more practical* (experts A and B).

Guideline 16: Rejected by experts due to practicality issues.

c) Added guidelines The new guideline included in this component is:

Guideline DC1: *"Ensure confidentiality does not hinder functionality by allowing queries to be processed on encrypted data"*. The guideline is added based on practicality concerns raised by the experts.

5.4.2.4 The Integration Approach Component

a) The quantitative results The results of applying one sample t-test to guidelines 18 to 22 are shown in Tables 5.11 and 5.12. The results show that the statistical significance $p < 0.05$; therefore, $H0$ is rejected. This means that the experts think the guidelines for this component are acceptable.

Table 5.11: One-Sample t-test for the Integration Approach Guidelines

Guideline	N	Mean	Std. Deviation	Std. Error Mean
18	14	4.5000	.85485	.22847
19	14	4.7143	.82542	.22060
20	14	4.5714	.51355	.13725
21	14	4.5714	.51355	.13725
22	14	4.4286	.64621	.17271

Table 5.12: One-Sample t-test for the Integration Approach Guidelines

	Test Value = 3.41					
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Guideline 18	4.771	13	<.001	1.09000	.5964	1.5836
Guideline 19	5.912	13	<.001	1.30429	.8277	1.7809
Guideline 20	8.462	13	<.001	1.16143	.8649	1.4579
Guideline 21	8.462	13	<.001	1.16143	.8649	1.4579
Guideline 22	5.898	13	<.001	1.01857	.6455	1.3917

b) The qualitative findings The guidelines of this component were changed as follows.

Guideline 20: Expert B suggest adding, *Link this process with the security policy*, as a detail of this guideline.

Guideline 22: Several suggestions were added to the details of this guideline, as follows:

- *Data can be classified as: highly classified, confidential, or public* (expert B).
- *The annotation needs to be in conjunction with an enforceable security policy* (expert F).

c) Added guidelines One guideline was added to this component:

Guideline IA1: *Monitor and log the process of passing the results of the query to the consumer (considering carrier precautions, encryption agreements, and transmission) (expert A).*

5.4.2.5 The Integration Location Component

a) The quantitative results The results of applying one sample t-test to guidelines 23 to 27 are shown in Tables 5.13 and 5.14. The results show that the statistical significance $p < 0.05$; therefore, $H0$ is rejected. This means that the experts think the guidelines for this component are acceptable.

Table 5.13: One-Sample t-test for the Integration Location Guidelines

Guideline	N	Mean	Std. Deviation	Std. Error Mean
23	14	4.6429	.84190	.22501
24	14	4.8571	.36314	.09705
25	14	4.5000	.75955	.20300
26	14	4.6429	.63332	.16926
27	14	4.9286	.26726	.07143

Table 5.14: One-Sample t-test for the Integration Location Guidelines

	Test Value = 3.41					
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Guideline 23	5.479	13	<.001	1.23286	.7468	1.7190
Guideline 24	14.911	13	<.001	1.44714	1.2375	1.6568
Guideline 25	5.369	13	<.001	1.09000	.6514	1.5286
Guideline 26	7.284	13	<.001	1.23286	.8672	1.5985
Guideline 27	21.260	13	<.001	1.51857	1.3643	1.6729

b) The qualitative findings The guidelines of this component were changed as follows.

Guideline 26: Expert F suggested the following detail: *Consider compliance of the service with known standards to establish trustworthiness, e.g. compliance with ISO 27000.*

Guideline 27: Expert I suggested clarifying the required protection by adding the following detail: *Protection is achieved by using a technique such as encryption.*

c) Added guidelines The following is the added guideline to this component:

Guideline IL1: *Monitor and log the data that is shipped in and out of this component*, (expert B).

5.4.2.6 System Security Management (SSM) Component

a) The quantitative results The results of applying one sample t-test to the guidelines 28 to 31 are shown in Tables 5.15 and 5.16. The results show that the statistical significance $p < 0.05$; therefore, $H0$ is rejected. This means that experts think the guidelines for this component are acceptable.

Table 5.15: One-Sample t-test for the SSM Guidelines

Guideline	N	Mean	Std. Deviation	Std.Error Mean
28	14	4.7143	.82542	.22060
29	14	4.2143	.89258	.23855
30	14	4.8571	.36314	.09705
31	14	4.4286	.93761	.25059

Table 5.16: One-Sample t-test for the SSM Guidelines

	Test Value = 3.41					
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Guideline 28	5.912	13	<.001	1.30429	.8277	1.7809
Guideline 29	3.372	13	.005	.80429	.2889	1.3196
Guideline 30	14.911	13	<.001	1.44714	1.2375	1.6568
Guideline 31	4.065	13	<.001	1.01857	.4772	1.5599

b) The qualitative findings The guidelines of this component were changed as follows.

Guideline 30: The following are the details added by experts:

- *Use real-time log analysis to detect abnormalities* (expert B).

- *Log authorised users who are currently viewing confidential information* (based on practicality discussion with expert B).

Guideline 31: This guideline was moved to the security policies component.

c) Added guidelines The following is the added guideline to this component:

Guideline SSM1: *Define a reporting mechanism for violations of confidentiality, privacy or trust* (expert A).

Guideline SSM2: *Employ different users and different roles to manage data access; consider internal employees and external consumers* (expert J).

Guideline SSM3: *Test the adopted access control model to prevent unauthorised access that leads to many threats* (expert E).

5.4.3 The Comprehensiveness of the SecureDIS

To assess the comprehensiveness of the SecureDIS, two aspects are used: the extent to which the basic components of DIS architecture with a middle layer are covered, and the extent to which the CPT properties are covered. The following explain each aspect.

The first aspect assesses the suitability of the proposed components to represent the basic components of a DIS with a middle layer. This is achieved through the qualitative and quantitative results to questions Q8, Q9.d, and Q10 of the expert reviews, which serve purposes P2 and part of P6.

Analysing the quantitative responses of the experts regarding the degree of suitability of SecureDIS components in Q9.d, the t-test results show that the statistical significance $p < 0.05$; therefore, $H0$ is rejected. This means that experts think the components are suitable, see Tables 5.17 and 5.18.

Table 5.17: One Sample t-test for SecureDIS Components' Suitability

	N	Mean	Std. Deviation	Std. Error Mean
Components' Suitability	14	4.2143	.97496	.26057

The collected comments on question Q7 discussed the comprehensiveness of SecureDIS. Although the question was not direct, half the experts considered SecureDIS to be comprehensive. Expert A's comment was: *"These DIS guidelines represent a holistic approach for the first time, in order to minimise data leakage threats"*. Expert I said, *"it is very specific and thorough, if the level of monitoring is possible, it is an ideal situation"*. Expert F commented, *"it is a fairly good practice"*, and *"proactive"* (expert

Table 5.18: One-sample t-test for SecureDIS Components' Suitability

	Test Value = 3.41					
	t	**df**	**Sig. (2-tailed)**	**Mean Difference**	**95% Confidence Interval of the Difference**	
					Lower	**Upper**
Components' Suitability	3.087	13	.009	.80429	.2414	1.3672

K). Expert B saw it as an internal standard to a DIS. Finally, Expert H said that SecureDIS *"looks very helpful and can be used"*.

Expert A suggested making the security policies and the System Security Management (SSM) part of every component of the DIS architecture, as they cannot be separated from the other components. This is a suitable suggestion if the focus is on a specific component of the system rather than the whole system. Expert D suggested that there is a need for a new component, responsible for handling the security and privacy of the system, which is basically the role of the security policies component.

The second aspect assesses the coverage of the CPT properties (expressed by questions Q9.a, Q9.b and Q9.c that serve purpose P2). The quantitative results show that for confidentiality, privacy and trust, the statistical significance $p < 0.05$; therefore, $H0$ is rejected. It means that the experts thought the coverage of CPT in SecureDIS is adequate. Tables 5.19 and 5.20 provide the statistical details.

Table 5.19: One-sample t-test for Confidentiality, Privacy, and Trust in SecureDIS

Guideline	**N**	**Mean**	**Std. Deviation**	**Std. Error Mean**
Confidentiality	14	4.2857	.82542	.22060
Privacy	14	4.3571	.92878	.24823
Trust	14	4.0714	.99725	.26653

Table 5.20: One-sample t-test of Confidentiality, Privacy and Trust in SecureDIS

	Test Value = 3.41					
	t	**df**	**Sig. (2-tailed)**	**Mean Difference**	**95% Confidence Interval of the Difference**	
					Lower	**Upper**
Confidentiality	3.970	13	.002	.87571	.3991	1.3523
Privacy	3.816	13	.002	.94714	.4109	1.4834
Trust	2.482	13	.028	.66143	.0856	1.2372

However, the qualitative analysis shows that several experts were concerned with *trust*. Expert C thought that if trust rights/models are related to legal aspects, they were out of scope of this study. Experts M and F thought that trust was vague terminology and

they were unsure how it could be implemented. Expert D suggested a periodic trust evaluation, while expert I suggested considering the impact of time and events on the trustworthiness of any entity. Expert D proposed basing trust models on the reputation of the entity and the general purposes of the entity in providing a service. According to expert D, consumers' trust can be checked by asking: Which organisation do consumers belong to?, What is the purpose of the query?, and What is the history of the consumer?

In terms of *confidentiality*, expert C suggested enforcing the confidentiality on: 1) consumers' logs, which contain queries, time, and date; 2) other logs that track query processing and system performance.

Based on the results of this theme, more details of trust and trust models needs to be provided to ensure improved coverage within the SecureDIS framework and guidelines.

5.4.4 The Practicality of SecureDIS

To assess whether is SecureDIS practical, several aspects need to be discussed: whether SecureDIS is acceptable as an approach, is implementable, is suitable for software engineers, and can be customised to a given context. Each of these aspects is now addressed.

The First Aspect: The Guidelines are Acceptable

The practicality of the proposed guidelines depends on whether or not guidelines as an approach are acceptable to be used to design and build secure systems. Discussing current practice within different organisations to mitigate data leakage threats helps to serve purpose P6. It is expressed by questions Q1, Q2, Q5 and Q7. The adopted practices can be one or a combination of the following:

- The use of international standards, such as: ISO 27000 (ISO, 2014) (Expert A and F), Orange book (expert H), NIST[4] (expert N), and COBIT 5[5] (Expert A).
- The use of government guidelines, for example, the UK government code of practice for open data for publishing health/crime data (expert D), a combination of the UK and European policies (expert N), and the ICT government policy (expert E).
- The use of other types of guidelines, such as organisation-specific technical guidelines (experts I and M), Data Management International (DAMA)[6](expert K), project-specific data security guidelines that can be a combination of several guidelines (expert K), Microsoft best practice in network security (expert C), Cloud

[4]NIST: National Institute of Standards and Technology, U.S. Department of Commerce. https://www.nist.gov/

[5]COBIT 5: a framework for the governance and management of enterprise IT, more information here http://www.isaca.org/cobit/pages/default.aspx?cid=1003566& appeal=pr

[6]DAMA International https://www.dama.org/

security frameworks and working practice in cloud security (expert N), and Digital Curation Centre (DCC)[7] guidelines, and Simple Property Oriented Threat (SPOT)[8] model for risk assessment (expert J).

Experts also discussed some approaches adopted to ensure security, such as:

- Basing data exposure on a need to know basis, using masking and scrambling techniques to protect the data, at the same time keeping its referential integrity (expert B).
- Using MD5 encoding in databases (expert L).
- Using SHA-1 cryptographic hash function (expert H).
- The incidence-driven proactive approach, when an event happens, the measures are considered (expert K).
- Limiting consumer queries by requesting administrative approval before any required query is processed (expert K).

The reasons for adopting a specific standard, set of guidelines, or best practice over others were discussed by experts.

- The selected approach is suitable for the organisation's needs across different layers of the organisation (experts L, G, K and J).
- It provides more technical details (expert C).
- It is used by other trusted companies (expert G).
- It is in compliance with government requirements (expert G).
- It is the lowest level demand for regulatory and financial requirements: selection of guidelines is based on the available budget to comply with them (expert E).
- It is derived from international standards (expert I).
- It is prestigious and reputable, regardless of the benefits (expert G).
- It provides basic security and compliance level (expert N).

These reasons show that security guidelines are one of the widely used approaches in practice to ensure security in systems development. Based on that, SecureDIS guidelines have the potential of being accepted in real data integration applications.

The Second Aspect: The Guidelines are Implementable

The qualitative analysis of the experts' responses to question Q7 shows that practicality is an issue (expert E). Several experts were concerned with the practicality of the guidelines in real applications (expert I), and the reasons for this concern are:

[7]Digital Curation Center http://www.dcc.ac.uk/

[8]The SPOT Model for Risk Assessment http://www.dlib.org/dlib/september12/vermaaten/09vermaaten.html

- Vagueness in defining the purposes for which the data are to be used (expert M). The concern is whether the purposes are understandable in an automated way.
- Difficulty in providing the purpose of using the data with every data request (experts H and I) and the difficulty of enforcing the matching of purposes between the data provider and the data consumers.
- Performance issues in tracking queries and profiling potential threats (experts H and I). According to them, it would take a long processing time for every data source to be added and every query to be processed.

In terms of *the purpose of using data sources*, expert M believed that if there is a controlled vocabulary that expresses different purposes, it will be reasonable and manageable. Although expert M regarded these guidelines as do-able, the expert warned that the performance of the system adopting them needs to be considered. Expert A suggested a solution to overcome performance issues and latency problems in resolving queries, which involves monitoring the requested queries and consumers' behaviour for a certain time. This would help in predicting the type of threats and therefore adopting specific prevention techniques.

In terms of *tracking queries*, expert A suggested a short behaviour analysis of the consumers' queries to develop a pattern. Since data is provided with a clear policy and consumers' roles are also clear, if consumers request more than they are allowed, the system could flag that kind of behaviour. Expert B, on the other hand, suggested analysing such patterns and creating rules accordingly, and limiting the process of monitoring behaviour to a certain time, to avoid unneeded processing that takes time and space. The experts suggest the use of statistical methods to sample consumers' behaviour or predict future behaviour towards violating countermeasures against data leakage.

The Third Aspect: The Guidelines are Suitable

Assessing whether the proposed guidelines are at an appropriate level to be understood and used by software engineers was addressed. Several responses to question Q7 highlight the suitability of the level of detail for software engineers. Although expert N thought that SecureDIS lacked sufficient detail to be used by a Chief Information Security Officer (CISO), expert C thought that the guidelines were at a non-technical level and therefore suitable for systems analysts. The remaining experts were asking about methods of implementation. This implies that the guidelines provided were generic, which targets software engineers' area of concern.

The Fourth Aspect: The Guidelines are Customisable

The qualitative analysis of the experts' comments on SecureDIS in question Q7, shows the architectural components and guidelines provided are indeed generic, although it

"has clear boundaries", as suggested by expert H. Several dimensions of customisation were suggested:

- The type of domain the DIS is created for. It can be for health, scientific research, business, etc. This customisation would have an effect on the SecureDIS components and guidelines (expert J). It would also affect the way security policies are defined, e.g. a research domain DIS has a policy of removing PII, while a business domain DIS may need to seek data owners' consent (expert E).
- The entity that a DIS is concerned to protect. Expert G raised the question: Is SecureDIS concerned with: 1) personal data, the data provided by data owners that needs to be analysed in terms of conflict of interest, and the issues of person vs. company needs to be resolved (expert I), 2) data providers, 3) the organisation building the DIS, 4) or data consumers?
- The level of implementation. Is SecureDIS designed for a web environment that uses HTTP (expert M)? Is it a semantic-oriented solution (expert H)?
- Type of data used in the DIS. Is the data structured, e.g. relational databases? or semi-structured, e.g. RDF (experts B, M, and J)?

5.4.5 Other Issues to Consider

This theme aims to gather experts' opinion of SecureDIS that were not directly part of the research questions, which achieves purpose P7. This could assist in shaping the future work of this study.

The time available to complete a data integration project is one of the factors that hinders enforcing security and privacy measures. Expert K discussed this point in the context of an urgent government project that required the use of referral to a higher authority to obtain data urgently. Therefore, there may not be enough time to enforce the analysis on the data before or during integration, in an ideal design. Security needs to be ongoing and proactive (expert B). Therefore, threat countermeasures need to be present, such as the use of exception handling as a method of secure coding, which was suggested by expert C, or the development of more advanced test cases for the ETL[9] process, as suggested by expert K.

In securing any system, human factors need to be considered (experts B and K). Expert L suggested having strict sanctions on deliberate human behaviour that threatens data leakage. Expert K suggested having proper feedback from the system to data consumers to make them aware of the policies and restrictions on the data. Users need to report any suspicious activity occurring in the system, especially when it looks very real, such

[9]ETL: Extract, Transform, Load

as email phishing (expert M) or when data quality is very bad (expert K), which could give an indication of other issues related to trust. Therefore, the culture of reporting a threat needs to be spread (expert G). All company employees need to be monitored and managed, especially new ones (expert A). As a last suggestion, expert H proposed limiting the sharing of login credentials to avoid data leakage to unauthorised entities.

Concerning *"What if security measures fail?"*, expert K asked, *"How does the system react in case of an attack, what if the design fails? What are the countermeasures for that?"* Expert B suggested no shutdown of the system while it is under attack. Access can be controlled more while monitoring the behaviour; at the same time, it is essential to contact authorities immediately to deal with the situation.

5.5 Discussion

One of the aspects behind this research is the observation by Clifton et al. (2004) that *"a comprehensive framework is necessary that handles the fundamental problems underlying privacy-preserving data integration and sharing"*. Another trigger to study data integration is the need to continue providing the integration of confidential data while maintaining the security and privacy requirements of the data.

The proposed guidelines have undergone refinement and have been categorised under each component to help investigate those components individually, while maintaining a vision of the whole system. This provides a broad perspective that a software engineer needs in order to build a secure system from the start. In addition, the proposed guidelines aim to cover the CPT properties, to gain another dimension of comprehensiveness that is also assumed to be useful to software engineers. The experts' confirmation of the proposed guidelines have shown that the framework is comprehensive and generally accepted.

The proposed guidelines were designed to mitigate data leakage threats. However, this was perceived by several experts to be impractical. If the guidelines are to be followed literally, it is probable that the system will experience poor performance since each data source and each query will have to be analysed, documented, profiled, logged and monitored. This is inefficient when timeliness is the most important property of the system. However, some experts thought that the guidelines are implementable, and there is a need to have clear strategies to overcome latency when answering consumers' queries and adding data sources. In addition, since the proposed framework and guidelines are generic, customisation can be a process that helps in overcoming practicality issues.

As security is the main focus of this investigation, so data leakage is among the most important threats to data integration. Studying data leakage threats in a data integration context is not a trivial task. This study began by collecting and investigating

data leakage threats occurring in a DIS context, alongside approaches that mitigate those threats present in the same context. The threats found in the literature are not seen as exhaustive; therefore, the experts' suggestions of additional threats are legitimate. Everyday new threats appear on the Internet, either to the infrastructure or to the resources. Several experts highlighted HTTP-level transmission threats, which were intentionally excluded from the data leakage threats as they are generic threats that can occur in any online system, and are not directly relevant to the integration process. However, more threats can be included and discussed in the context of DIS if the focus is to span all levels of the OSI model.

The experts had a legitimate concern about the notion of risk. It is important to assess the threats' degree of severity before preventing them. However, risk assessments and a detailed threat analysis require a specified running system and information about its security policies. The assessment would involve a thorough data collection, including but not limited to: services running, network assessment, access control permission (Bayne, 2002). This is out of the scope of this study, which aims to investigate the conceptual system and the possible leakages in general. Chapter 4 attempted a threat analysis based on the conceptual architecture, sufficient for this study.

The experts selected varied in their level of technical expertise; some had management and policy level experience, while some had technical level. Despite these differences, the experts agreed on the importance of security policies in DIS. Those with a high-level perspective focused on how natural language security policies could be implemented, while experts with technical expertise visualised security policies as access control rules. Based on these views, and the work proposed by Haddad et al. (2012), the integration process needs to be guided by a global policy. This approach ensures that none of the security policies for the entities involved is neglected, and can therefore be maintained.

Choosing a DIS architecture with a middle layer helped in determining the scope of this research. The study of data leakage within that architecture, as presented in Section 4.2.7, emphasised the location of leakages within the architecture. This helped in categorising the threats by DIS component.

Discussing with the experts current practice used by organisations to protect themselves against data leakage was informative. Most experts use guidelines put together by their organisation, or represent best practice in the organisation's domain. Only a few were using internationally-known standards, such as ISO 27000 (ISO, 2014). It appears that developers are not fully aware of the guidelines used by their organisation. This is an indicator of the disconnection between the management and the employees, which in turn affects how security is implemented and maintained. The importance of having implementable guidelines was demonstrated by expert K who had never analysed data to remove QIDs, which explains why inference attacks had occurred in expert K's organisation. However, the same organisation was somewhat successful in limiting

consumer-side data leakage, by not allowing direct queries to its system. It has chosen formal request letters, signed by authorised employees, as a method for querying their system, which thus limits consecutive query attacks.

Trust models were raised by several experts. The criteria resulting from the discussion of trustworthiness in data sources (see Section 5.4.2.1) can be used with the entities of the system to determine their level of trustworthiness.

Human factors are also important. The disconnect between developers and policy-makers explains how security can be compromised. Security policies need to be manageable and implementable, otherwise there is no need to create them. In addition, employees need to be educated in these policies and the consequences of their violation.

The qualitative and quantitative analysis of the expert reviews, in Chapter 4 and in this chapter, demonstrated the need for more customisation in later chapters. SecureDIS can be used by software engineers to develop security policies as discussed in Chapter 6. However, to find out its practical use, SecureDIS needs to focus on a specific domain, so Chapter 7 presents one application to the healthcare domain.

5.6 The Confirmed SecureDIS Framework and Guidelines

As a result of the experts' reviews, changes were made to SecureDIS. These include: modifications to the existing guidelines, newly added guidelines, deleted guidelines. Tables 5.21 to 5.26 show the synthesis of the changes made to the guidelines in each component linked with data leakage threats.

5.7 Use of SecureDIS by Software Engineers

Organisations developing their own DIS need to ensure the security and privacy of the system especially when handling sensitive data. Security guidelines in general can be used during the analysis and early design phases of systems development (Khan and Zulkernine, 2009). Hence, SecureDIS can be used in a similar fashion. Software engineers can benefit from using SecureDIS framework and guidelines by the following:

- Since the guidelines are organised under each component of the DIS architecture, software engineers can focus on a particular component to ensure that the CPT properties and countermeasures against data leakage are covered during development.
- The guidelines can be tailored according to the organisation's needs and budget. Hence, they can be extended, modified, or removed, depending on the context of

the data integration and the required level of security. This is similar to many existing security guidelines, such as the software security checklist provided by Gilliam et al. (2003).

- The guidelines can be converted into a qualitative evaluation checklist to compare and evaluate the security level of several vendor-provided data integration products and can therefore help to choose a suitable product for an organisation.
- The guidelines can be used to build the security policies of a DIS by ensuring the coverage of the CPT properties by each component. An example of this application is discussed in Chapter 6.

5.7.1 SecureDIS and Other Software Engineering Approaches

The literature addressing secure data integration is summarised in Table 2.3 and compared with the elements of the scope of this study. However, the LINDDUN approach, see Section 2.4.2, is found to be the most relevant approach to SecureDIS from the security and software engineering side rather than from the data integration side. The methodology is similar to SecureDIS in covering confidentiality and privacy properties and their techniques; however, LINDDUN does not consider trust. LINDDUN targets software engineers to assist them in eliciting security requirements; but it is generic, and does not focus on the data integration context nor on the use of security policies. The details of the comparison between LINDDUN and SecureDIS is shown in Table 5.27.

5.8 Summary

This chapter introduced SecureDIS, a framework that aims to assist software engineers in designing a secure DIS that mitigates against data leakage by considering the CPT properties. The complex nature of DIS requires investigating the system from an architectural perspective to simplify the solutions proposed. SecureDIS represents the architectural components of a DIS, which are: data and data sources, security policies, data consumers, the integration approach, the integration location, and System Security Management (SSM). Each component contains a set of guidelines to mitigate data leakage in that component. Each guideline focuses on one or more CPT property against one or more data leakage threats. The guidelines presented for the SecureDIS can be used by software engineers for building DIS, for creating security policies, or for evaluating existing DIS. SecureDIS addresses the research question RQ2, see Section 3.1, that aims to assist software engineers in mitigating data leakage threat in DIS.

The SecureDIS guidelines required further improvements and evaluation; hence a second experts' reviews was carried out. In terms of the *validation of the guidelines*, the

Table 5.21: Confirmed Guidelines for the Data and Data Sources Component

No.	Prev. No.	Guideline	DL Threat	C	P	T
1	**1**	Check for security meta-data	DL19	✓	✓	
2	**2**	Check for privacy requirements that contain the purpose statement of the used personal data	DL19	✓		
3	**3**	Based on the context of the DIS, remove attributes that can directly identify a person and add proper identity information that does not harm the privacy of an individual	DL25		✓	
4	**4**	Analyse data to replace/remove attributes, such as QIDs, that have potential for inference by correlation or computation, using appropriate techniques	DL7, DL8, DL9		✓	
5	**5**	Create a security policy for each data source, based on security meta-data and privacy requirements, including the privacy loss tolerated	DL19	✓	✓	
6	**6**	Exclude sensitive data that can be recognised and inferred by deduction	DL21		✓	
7	**DS1**	Determine the trustworthiness of the data sources by monitoring different aspects (e.g. entity type, update frequency, technical properties)	DL22, DL23			✓
8	**DS2**	Provide feedback to data provider about the use of personal data including: the purpose of using data, the entity that used the data (or the role name), the part of the data that was used, and the date and time of use	DL24		✓	

experts' responses show statistically that the guidelines were accepted. The responses collected about each guideline introduced some minor changes to the existing guidelines and provided more details for some of them. The experts also contributed to the framework by proposing 10 additional guidelines. This expands the framework from 31 to 41 guidelines.

Table 5.22: Confirmed Guidelines for the Security Policies Component

No.	Prev. No.	Guideline	DL Threat	C	P	T
9	**7**	Ensure the security policy considers the integrated security policies of the participating data sources, the DIS itself, the integration platform, and the data protection law for each country involved in the integration	DL15, DL16, DL17	✓	✓	
10	**8**	Define third parties' and cloud services providers' (public, private, hybrid, and community) rights on data (including transitive trust)	DL11, DL12, DL13	✓	✓	✓
11	**31**	Where using cloud services, or third parties to handle data (i.e. to access, process, store, or manage data): establish the required trust to achieve those tasks	DL11, DL12, DL13	✓	✓	✓
12	**9**	Define data providers' rights to access other data within the DIS	DL20	✓	✓	✓
13	**10**	Define trust models used with system components and users	DL22, DL4			✓
14	**11**	Ensure data access and sharing is based on matching the purpose statement of both the data consumer and the data sources' privacy requirements	DL4		✓	
15	**12**	Define data consumers' consumption rights, considering trust as one parameter that changes data access	DL4	✓		✓
16	**SP1**	Define how the compliance to policy or lack of it is enforced	DL24	✓	✓	✓
17	**SP2**	Test the proposed security policies before implementation	DL14, DL1, Dl2, Dl3	✓	✓	✓
18	**SP3**	Consider the risks associated with the violation of security policies and trust model	DL24	✓	✓	✓

The quantitative results show that the SecureDIS framework is considered *comprehensive* in terms of its coverage of CPT properties. This result is well-matched with the

Table 5.23: Confirmed Guidelines for the Data Consumers Component

No.	Prev. No.	Guideline	DL Threat	C	P	T
19	**13**	Attempt resolving queries only when after access to data is granted	DL1, DL2, DL3	✓		
20	**14**	Analyse the features of the query, e.g. type of predicates and types of data returned, to identify possible security and privacy breaches and to conform to the security policy	DL7, DL9	✓	✓	
21	**15**	Keep record of all queries and classify them according to 'type of threat' to prevent consecutive queries inference attacks	DL6		✓	
NA	**16**	Rewrite the queries after applying authorisation rules, privacy policies, and metadata restrictions	DL18	✓	✓	
22	**17**	Protect query information (such as location and predicates)	DL5		✓	
23	**DC1**	Ensure confidentiality does not hinder functionality by allowing queries to be processed on encrypted data	DL10	✓		

qualitative findings that show approval of the comprehensiveness of SecureDIS in both the coverage of the CPT and its architectural components.

For the *practicality* of the proposed guidelines, the qualitative findings were categorised into four aspects. The first was whether the guidelines are acceptable as a method to mitigate data leakage. The second aspect was whether the guidelines are implementable. The third was the suitability of the level of detail for software engineers. The fourth aspect was the degree to which the guidelines are generic but can be customised to a given domain. The findings showed that, in the opinion of these experts, the guidelines are acceptable, implementable, suitable, and customisable. More customisation details are needed to use SecureDIS for a real data integration application aimed at data leakage mitigation.

Other collected observations were useful in reshaping the guidelines. One was the attempt to find criteria that help assess the trustworthiness of an entity. However, this process needs to start by defining a trust model and how it can be implemented within

Table 5.24: Confirmed Guidelines for the Integration Approach Component

No.	Prev. No.	Guideline	DL Threat	C	P	T
24	**18**	Before the integration: Select suitable privacy-preserving data integration techniques based on query analysis	DL7, DL8, DL9		✓	
25	**19**	During the integration: encrypt data, in general, against the platform (i.e. integration location)	DL10	✓		
26	**20**	After the integration: analyse the query results to predict possible security and privacy policy violations and to determine further techniques to be applied before the result is returned to the consumer	DL7, Dl8, DL9, DL25	✓	✓	
27	**21**	After the integration: compute the aggregated privacy loss of the integrated results using the measure of privacy loss tolerated from each data source to satisfy privacy requirements	DL18		✓	
28	**22**	After the integration: annotate the results of the query with security and privacy meta-data and provide a clear description of their contents	DL19	✓	✓	
29	**IA1**	Monitor and log the process of passing the results of the query to the consumer (considering carrier precautions, encryption agreements, and transmission)	DL24	✓	✓	✓

the system. It continues by deciding when and how frequently trust is evaluated, in addition to understanding the effect of untrustworthy events on the overall trustworthiness of an entity.

The chapter discussed several data leakage threats including consumer-side inference attacks and vendor attacks. Further, the suitability of the architectural components of SecureDIS was discussed. Analysis of the results shows that the components are considered suitable. However, the qualitative part of the analysis provided only a few suggestions for improvements, which were not necessarily viable.

Table 5.25: Confirmed Guidelines for the Integration Location Component

No.	Prev. No.	Guideline	DL Threat	C	P	T
30	**23**	Comply with security and privacy policy to prevent data leakage	DL18	✓	✓	✓
31	**24**	Obtain the data licences needed to access data sources	DL18, DL19		✓	
32	**25**	Establish trust with data providers who provide data sources	DL22			✓
33	**26**	In the case of outsourcing to an integration service, ensure the trustworthiness of the integration method/service to integrate the data	DL11	✓	✓	✓
34	**27**	Protect the confidentiality of the data before and after the integration is achieved, (within the layers of the integration location)	DL10	✓		
35	**IL1**	Monitor and log the data that is shipped in and out of this component	DL24	✓	✓	✓

SecureDIS is different from current approaches in several aspects. It covers the data integration context and the components needed in a DIS architecture. It focuses on the CPT properties and considers the security and privacy policies of data sources provided by different organisations. Its main concern is to mitigate data leakage in a DIS context. Table 5.27 summarises the comparison between SecureDIS and LINDDUN, a closely-related approach.

Table 5.26: Confirmed Guidelines of the SSM Component

No.	**Prev. No.**	**Guideline**	**DL Threat**	**C**	**P**	**T**
36	**28**	Conduct audits to ensure compliance with the DIS's security policies (including confidentiality, privacy, and trust)	DL15, DL16, DL17, DL24	✓	✓	✓
37	**SSM1**	Define a reporting mechanism for violations of confidentiality, privacy, or trust	DL24	✓	✓	✓
38	**29**	Configure the access control model used to include privacy needs	DL1, DL2, DL3	✓	✓	
39	**SSM2**	Employ different users and different roles to manage data access; consider internal employees and external consumers	DL1, DL2, DL3	✓		
40	**SSM3**	Test the adopted access control model to prevent unauthorised access	DL1, DL2, DL3, DL24	✓	✓	✓
41	**30**	Monitor and log successful and unsuccessful access to data (especially private data)	DL24	✓	✓	✓

Table 5.27: Comparison between LINDDUN and SecureDIS

	LINDDUN	**SecureDIS**
Approach	A methodology	A set of guidelines
Purpose	To elicit privacy requirements	To build secure DIS that mitigate the threats of data leakage caused by lack of considering CPT in early system design
Based on	STRIDE	Literature review and the DIS architecture
Target Audience	Software engineers	Software engineers
Nature	Contains steps to consider privacy and provides suggestions on privacy techniques	Natural language guidelines linked to CPT properties
Context	Generic	Specific to the data integration context
Architecture	Generic	Data integration systems with middle layers
Threats	Based on the DFD and the misuse cases, threat trees are constructed	Based on the data leakage threats identified in the literature and categorised by DIS component location and CPT property
Covered properties	Linkability, identifiability, non-repudiation, detectability, disclosure of information, content unawareness, policy and consent non-compliance	Confidentiality, privacy, and trust
Confidentiality elements	Access control, data protection	Access control, data protection
Privacy elements	All covered properties are privacy elements	Anonymity, compliance with data use policies and regulatory requirements
Trust elements	Trust is not considered	Behaving according to policy, trustworthiness assessment, adopting a trust model
SDLC Phase	Links to early development stages of SDLC	Links to early development stages of SDLC
Data Oriented	Yes, based on DFD	Yes, based on resolving users' queries that use multiple data sources
Emphasis on security policy	N/A	Yes, multiple data sources and entities with multiple security policies

Chapter 6

Modelling DIS Security Policies

Emerging applications, such as DIS, introduce new security challenges and therefore would benefit from the features of formal methods in capturing precise rules against defined threats. The security policies of DIS can benefit from formal methods by verification using model checkers.

SecureDIS emphasises the importance of considering the CPT properties to mitigate data leakage threats, as discussed in Section 4.4. These properties can be used as elements of the security policies, specified and enforced by the DIS, to mitigate data leakage. This chapter discusses how the Event-B formal method, supported by the Rodin toolset, was used to ensure the correct specification of the DIS security policies. Through abstraction and refinement, the model targets specific elements that belong to each property of the CPT and converts them into rules implemented by Event-B.

The chapter starts by an overview of the Event-B formal method. It discusses the system requirements derived from SecureDIS guidelines that will be used to create the security policies. The chapter then explains the modelling process and discusses the evaluation of the model. Towards the end of the chapter, a reflection on the SecureDIS based on the model is presented, and a summary of the main findings is explained.

6.1 Overview of the Event-B Formal Method

A brief description of Event-B as a formal method was presented in Section 3.3.3. An Event-B model consists of two main components: CONTEXT and MACHINE. The CONTEXT includes the static part of the model that defines SETS, CONSTANTS, and AXIOMS that add constraints to the sets. The MACHINE contains the dynamic part of the model that includes VARIABLES, INVARIANTS, and EVENTS. The VARIABLES specify the states of the system and can be modified by *guarded* EVENTS. The INVARIANTS specify the constraints on variables, which need to be proved true at every state

of the system. The verification of the model demonstrates consistency by ensuring the correctness among all refinement levels.

The integrated toolset used to model Event-B is Rodin (Abrial et al., 2006). The verification process achieved by Rodin includes: 1) Model checking: by ProB (Leuschel and Butler, 2003) a model checker integrated in Rodin, and 2) Theorem proving: by generating and proving proof obligations.

6.2 SecureDIS Guidelines and Requirements for Security Policies

Modelling DIS security policies using formal methods is beneficial because: 1) the CPT properties can be precisely captured, 2) the modelling process is presented to help system designers verify security policies before implementation, which encourages the concept of security by design, 3) the correctness and consistency mitigates data leakage threats related directly to the CPT elements covered in this model.

This model focuses on several SecureDIS guidelines related to the specification and enforcement of security policies achieved at the integration location. The security policies capture the CPT properties discussed in Section 4.4. Existing studies focus on one or two properties of the CPT and do not consider the type of system being targeted here, see Section 2.4.5. The policies include the following elements:

- For confidentiality, the model includes the use of RBAC model as an approach to avoid unauthorised access. RBAC was chosen as it is capable of modelling a wide range of access control policies (Shafiq et al., 2005).
- For privacy, data classification and data use purposes are the elements chosen to be modelled, as techniques to counter non-compliance with data sources' policies.
- A trust model is adopted to estimate the risks caused by authorised consumers, to be considered during data access. It is worth mentioning that there is a lack of studies in the area of security policies modelling using Event-B that capture trust elements.

Table 6.1 shows how the SecureDIS guidelines are transformed into specific system requirements, where the Property column indicates the CPT property covered by the requirement.

Table 6.1: System Requirements Details

Guideline	Req. No.	System Requirement	Property	Type
39	1	Each data consumer must be assigned to a role to access data sources items	C	Specification
1	2	Each data source specifies which roles are allowed to access the data sources items	C	Specification
1, 19	3	A data consumer is granted access to data items returned by a query if the assigned role is an allowed role	C	Enforcement
14	4	Each data consumer specifies a purpose to access data items	P	Specification
2, 14	5	Each data item is associated with a purpose for which it was collected	P	Specification
14	6	A data consumer is granted access to data items returned by a query, if the purpose of the query matches the purpose for which the data items were collected	P	Enforcement
2	7	Each data item is classified based on its sensitivity	P	Specification
-	8	Each data consumer is assigned to a security level, which specifies the authorisation to access data of a certain sensitivity	P	Enforcement
-	9	A data consumer is granted access to data items returned by a query, if the security level of the consumer is equal to the sensitivity level of the data items	P	Enforcement
-	10	Each data consumer is assigned to a trust level	T	Specification
-	11	Data sources determine the acceptable data consumers trust levels	T	Specification
15	12	A data consumer is granted access to data items returned by a query, if the trust level of the consumer matches the accepted trust level of data items	T	Enforcement

6.3 Modelling Security Policies

A security policy consists of the following basic components: a subject, permission(s), and an object (Crampton and Huth, 2010), and targets a specific property. The security policies modelled in this chapter are derived from the system requirements in Table 6.1 that are focused on the CPT properties.

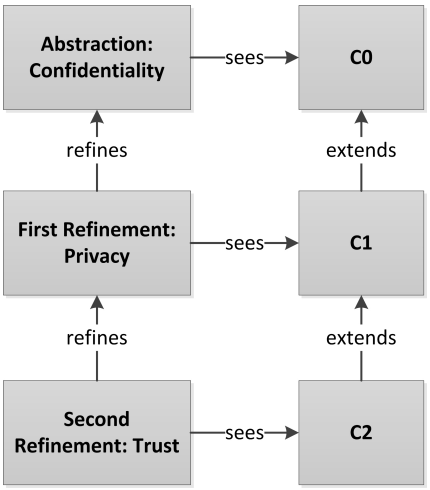

Figure 6.1: Refinements to Security Policies

The security policies are modelled through Event-B refinements to increase the complexity of the policies with each refinement. Three levels of refinement are proposed (see Figure 6.1), each level being represented by a CONTEXT: C0, C1, or C2.

1. **System abstraction**: captures the process of data consumers querying the data provided by different data sources in addition to the security policy that grants the execution of the query. Execution is only granted if the consumer is assigned to a specific role that provides permission to execute the query.
2. **The first refinement**: extends the security policy to include the purpose for which data items can be accessed in addition to the data sensitivity.
3. **The second refinement**: extends the security policy to include the trust levels that data sources should place in data consumers for granting them the execution of a query.

6.3.1 System Abstraction: Modelling Confidentiality

The first step is to model data consumers queries to different data sources and the RBAC policy governing query execution granted to consumers. The system abstraction includes four main sets: *DATA_CONSUMER*, the set of data consumers; *CONSUMER_ROLE*, the set of roles that can be assigned to consumers; *DATA_ITEM*, the set of data items associated with data sources and also returned by queries; and *DATA_SOURCE*, the set of data sources providing the data items to answer consumers queries.

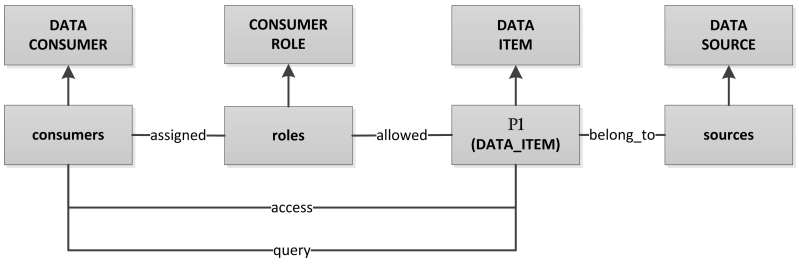

Figure 6.2: System Abstraction: Modelling Data Query and Confidentiality

The system abstraction also includes the main VARIABLES and EVENTS that capture the DIS environment, see Figure 6.2. The events are summarised as follows.

- ***AddDataSources***: to add data sources to the model.
- ***AddDataItemsToSources***: to create data items and associate them with data sources.
- ***AddDataConsumers***: to add data consumers to the model.
- ***AddRoles***: to add consumers' roles to the model.
- ***AssignRolesToConsumers***: to assign specific consumer roles to specific data consumers.
- ***AddConsumersQueries***: to create consumer queries containing data items.

The variable *belong_to* associates data items with their data sources, where multiple data items belong to multiple data sources. The invariant that ensures this relation is defined is:

$$\textbf{inv1 :} \quad belong\_to \in \mathbb{P}1(DATA\_ITEM) \leftrightarrow sources$$

Data consumers can access data items coming from data sources by creating a *query*. The variable *query* is defined as the relationship between consumers and data items. The following invariant show that multiple consumers can query multiple data items:

$$\textbf{inv2 :} \quad query \in consumers \leftrightarrow \mathbb{P}1(DATA\_ITEM)$$

However, the *query* has one main restriction that is when a consumer (c) requests a set of data items ($items$), these items need to belong to existing data sources (s). This restriction is enforced by the following invariant:

$$\textbf{inv3 :} \quad \forall c, items.c \mapsto items \in query \Rightarrow (\exists s.belong\_to\big[\{items\}\big] = s)$$

The *query* is created in the ***AddConsumersQueries*** event shown below. The event contains a list of parameters (ANY), a collection of guards (WHERE), and collection of actions (THEN). An event can execute its actions only when its guards are true. In this case, the event essentially needs to check whether data items map to sources in **grd4** to satisfy **inv3**.

Event *AddConsumersQueries*

ANY

$consumer, data\_items, source$

WHERE

grd1 : $consumer \in consumers$

grd2 : $(data\_items \in \mathbb{P}1(DATA\_ITEM)) \wedge$
$(data\_items \neq \emptyset)$

grd3 : $(source \in sources)$

grd4 : $data\_items \mapsto source \in belong\_to$

THEN

act1 : $query := query \cup$
$\{consumer \mapsto data\_items\}$

END

To specify the security policy that captures the confidentiality property, the following components are modelled.

- The *assigned* invariant denotes that data consumers can be assigned to more than one role, which fulfils sys. req. 1, and that a role can be assigned to one or more consumers, as follows:

$$\textbf{inv4 :} \quad assigned \in consumers \leftrightarrow roles$$

- The *allowed* invariant indicates the roles allowed to access the data items, which acts as the access control list. Also, *allowed* ensures that data items are actually coming from existing data sources (sys. req. 2). Both of these aspects are modelled as follows.

$$\textbf{inv5 :} \quad allowed \in roles \leftrightarrow \mathbb{P}1(DATA\_ITEM)$$
$$\textbf{inv6 :} \quad \forall role, items.role \mapsto items \in allowed \Rightarrow (\exists source.items \mapsto source \in belong\_to)$$

- The event ***AddAuthorisation*** adds the RBAC policy to the system by updating the variable *allowed*. To add the pair of data item (i) and role (r) to the allowed access control list, the guard **grd3** checks whether the data item is associated with existing data sources as follows:

Event *AddAuthorisation*
ANY
r, i, s
WHERE
grd1 : $i \in \mathbb{P}1(DATA\_ITEM)$
grd2 : $(s \in sources) \wedge (sources \neq \emptyset)$
grd3 : $i \mapsto s \in belong\_to$
grd4 : $(r \in roles) \wedge (roles \notin \emptyset)$
grd5 : $r \mapsto i \notin allowed$
THEN
act1 : $allowed := allowed \cup \{r \mapsto i\}$
END

To model the enforcement of the security policy specified earlier, the following is included:

- Two invariants: **inv7** to model the actual access of consumers to data items, and **inv8** to ensure the accessed items are returned by a query. Thus:

$$\textbf{inv7 :} \quad access \in consumers \leftrightarrow \mathbb{P}1(DATA\_ITEM)$$
$$\textbf{inv8 :} \quad \forall c, items.c \mapsto items \in access \Rightarrow (c \mapsto items \in query)$$

- The ***AccessData*** event checks whether the consumer is assigned to a role (**grd3**), and whether the assigned role is entitled to execute the query (**grd4**), to fulfil sys. req. 3. It also ensures the data items accessed by the consumer are returned as results of a query by the same consumer (**grd2**). The event is modelled as follows.

Event ***AccessData***
ANY
$consumer, data\_items,$
$consumer\_roles$
WHERE
grd1 : $consumer \in consumers$
grd2 : $data\_items \in query[\{consumer\}]$
grd3 : $(consumer\_roles \subseteq roles) \wedge$
$(assigned[\{consumer\}] = consumer\_roles)$
grd4 : $\exists role.(roles \in consumer\_roles) \wedge$
$(role \mapsto data\_items \in allowed)$
grd5 : $(consumer \mapsto data\_items) \notin access$
THEN
act1 : $access := access \cup \{consumer \mapsto data\_items\}$
END

6.3.2 First Refinement: Modelling Privacy

The system abstraction in Section 6.3.1 models who can have access to the data items returned by a query. This refinement extends the previous level by adding the privacy dimensions (purpose and data sensitivity) to the security policy, as discussed in the beginning of Section 6.3.

To model the *purpose*, the following components are introduced.

- A set $DATA\_USE\_PURPOSE$ is defined in the context (C1) to include all the possible data use purposes that can be assigned to data consumers or data items:

axm1 : $partition(DATA\_USE\_PURPOSE, \{research\},$
$\{commercial\}, \{personal\}, \{public\})$

- A variable *item_purpose* represents the relationship between the data items and their possible purposes.

$$\textbf{inv9 :} \quad item\_purpose \in \mathbb{P}1(DATA\_ITEM) \leftrightarrow DATA\_USE\_PURPOSE$$

- The variable *query_purpose* represents the relationship between the *consumers* and the data use purposes.

$$\textbf{inv10 :} \quad query\_purpose \in consumers \rightarrow DATA\_USE\_PURPOSE$$

- A new event, ***AddItemsPurposes***, assigns several purposes to data items (sys. req. 5). The guards and actions of the event ***AddItemsPurposes*** are as follows:

$$\begin{aligned} \textbf{grd1 :} \quad & purpose \in DATA\_USE\_PURPOSE \\ \textbf{grd2 :} \quad & i \in \mathbb{P}1(DATA\_ITEM) \\ \textbf{act1 :} \quad & item\_purpose := item\_purpose \cup \{i \mapsto purpose\} \end{aligned}$$

- The event ***AddConsumersQueries*** is refined to assign a purpose to each consumer request to query the system (sys. req. 4) by adding the following:

$$\begin{aligned} \textbf{grd1 :} \quad & purpose \in DATA\_USE\_PURPOSE \\ \textbf{grd2 :} \quad & c \in consumers \\ \textbf{act1 :} \quad & query\_purpose := query\_purpose \cup \\ & \{c \mapsto purpose\} \end{aligned}$$

To model the *data classification*, the following are included:

- A set named *CLASSIFICATION* is defined in the context (C1) containing the possible levels that can be assigned to data items and data consumers. The set includes the following labels.

  Regulated: data items that are protected by data protection regulations. For example, the items that contain the PII, such as names, SSN, and credit card numbers. If these items were disclosed, harm would be caused to the reputation of the data sources and may lead to financial losses.

Confidential: data items that include sensitive information that if disclosed could result in a medium level of harm and financial losses.

Public: data items that can be disclosed to the general public that if disclosed could result in a low risk to privacy and reputation.

$$\textbf{axm2 :} \quad partition(CLASSIFICATION, \{Regulated\}, \{Confidential\}, \{Public\})$$

- A variable *classified* links each data item with a *CLASSIFICATION* (sys. req. 7) as follows:

$$\textbf{inv11 :} \quad classified \in \mathbb{P}1(DATA\_ITEM) \nrightarrow CLASSIFICATION$$

- The event ***AddDataItemsToSource*** is refined to classify each data item by updating the variable *classified* as follows:

$$\begin{aligned} \textbf{grd1 :} &\quad i \in \mathbb{P}1(DATA\_ITEM) \\ \textbf{grd2 :} &\quad j \in CLASSIFICATION \\ \textbf{grd3 :} &\quad i \notin dom(classified) \\ \textbf{act1 :} &\quad classified := classified \cup \{i \mapsto j\} \end{aligned}$$

- A variable *security_clearance* associates a consumer with a security clearance. It is defined as follows:

$$\textbf{inv12 :} \quad security\_clearance \in consumers \rightarrow CLASSIFICATION$$

- The event ***AddDataConsumers*** is refined to assign each new data consumer an appropriate security clearance (sys. req. 8):

$$\begin{aligned} \textbf{grd1 :} &\quad sc \in CLASSIFICATION \\ \textbf{grd2 :} &\quad c \in consumers \\ \textbf{grd3 :} &\quad c \mapsto sc \notin security\_clearance \\ \textbf{act1 :} &\quad security\_clearance := security\_clearance \cup \{c \mapsto sc\} \end{aligned}$$

To enforce the extended security policy, the ***AccessData*** event is refined by including the following guards:

- A guard to enforce accessing data when the data consumer's purpose, during query creation, matches one of the data items purposes (sys. req. 6):

$$\textbf{grd6:} \quad item\_purpose[\{data\_items\}] = query\_purpose[\{consumer\}]$$

- A guard to ensure that the classification of data items requested for access matches the consumer's security clearance (sys. req. 9):

$$\textbf{grd7:} \quad security\_clearance[\{consumer\}] = classified[\{data\_items\}]$$

6.3.3 Second Refinement: Modelling Trust

The second refinement extends the first to capture the trust property. Trust is introduced into the security policy to minimise threats that are related to secondary disclosure of information caused by data consumers' abuse of privileges. This refinement introduces the trust model proposed by Agudo et al. (2010), which labels an entity with: very good, good, neutral, bad, and very bad, based on calculations conducted on that entity to assess its risks. This trust model is included in the second refinement by adding the following components:

- A set *TRUST_LEVEL* containing all possible trust levels in the trust model:

$$\textbf{axm3:} \quad partition(TRUST\_LEVEL, \{very\_good\}, \{good\}, \{neutral\}, \{bad\}, \{very\_bad\})$$

- A variable *consumer_tlevel* to associate each data consumer with its trust level:

$$\textbf{inv13:} \quad consumer\_tlevel \in consumers \rightarrow TRUST\_LEVEL$$

- A variable *item_tlevel* to associate data items with their acceptable trust levels:

$$\textbf{inv14 :} \quad item\_tlevel \in \mathbb{P}1(DATA\_ITEM) \leftrightarrow TRUST\_LEVEL$$

- The event ***AddConsumers*** is refined to associate a data consumer with its trust level during the addition of the consumer to the system (sys. req. 10):

$$\begin{aligned} \textbf{grd5 :} \quad & c \in consumers \\ \textbf{grd6 :} \quad & t \in TRUST\_LEVEL \\ \textbf{act3 :} \quad & consumer\_tlevel = consumer\_tlevel \cup \{c \mapsto t\} \end{aligned}$$

- The event ***AddDataItemsToSources*** is refined to associate data items with acceptable trust levels (sys. req. 11):

$$\begin{aligned} \textbf{grd5 :} \quad & i \in \mathbb{P}1(DATA\_ITEM) \\ \textbf{grd6 :} \quad & t \in TRUST\_LEVEL \\ \textbf{act3 :} \quad & item\_tlevel = item\_tlevel \cup \{i \mapsto t\} \end{aligned}$$

To enforce the security policy related to the trust property, the ***AccessData*** event is refined to check whether the consumer's trust level matches the expected trust level associated with the data items returned by a query (sys. req. 12). The following guard is included:

$$\textbf{grd8:} \quad item\_tlevel[\{data\_items\}] = consumer\_tlevel[\{consumer\}]$$

6.4 Formal Verification of the Model

The Rodin toolset provides an environment for both modelling and proving, by a) theorem proving and b) model checking. In addition to formal modelling, the proposed Event-B model is proved to be correct and consistent. Table 6.2 presents an overview of the proof efforts provided by Rodin. These statistics measure the Proof Obligations (PO) generated and discharged by the Rodin prover and the POs that are interactively proved. The complete development of the DIS security policies results in 38 POs, in which (100%) are proved automatically by Rodin. The number of POs in the system abstraction that captures the confidentiality property is larger than other refinements. This because the main components of the security policies (the subject, the permissions, and the object) have to be established, and therefore many invariants are introduced in that layer to guarantee the correctness of these components.

a) Theorem Proving: Different POs are generated by Rodin during the development of a system (Hallerstede, 2011). An example of a PO is demonstrated by the "Invariant

Table 6.2: The Statistics of the Model

Element Name	Total	Auto	Manual
Model	38	38	0
Confidentiality	25	25	0
Privacy	9	9	0
Trust	4	4	0

Preservation". The INV PO ensures that each invariant is preserved by each event. To prove that *inv6*, below, is preserved by the *AddAuthorisation* event, "AddAuthorisation/inv6/INV" PO is generated and proved by Rodin.

$$\textbf{inv6}: \quad \forall role, items.role \mapsto items \in allowed \Rightarrow (\exists source.items \mapsto source \in belong\_to)$$

To prove this PO, guard **grd3** below is added to the *AddAuthorisation* event to ensure that each role is linked to a data item *(i)* that actually belongs to a data source *(s)*.

$$\textbf{grd3}: \quad i \mapsto s \in belong\_to$$

b) Model Checking: ProB is an animator and a model checker for Event-B. ProB allows fully automatic exploration of Event-B models, and can be used to systematically check a specification for a range of errors. This model was analysed using ProB to ensure that the model is deadlock free. Each new event added to the refinements was verified as not introducing a deadlock.

6.5 Reflection on SecureDIS

The modelling of security policies addresses the part the research question RQ2 in Section 3.1 concerned with helping software engineers overcome data leakage threats in DIS by design. Selected SecureDIS guidelines are implementable by demonstrating how they can be transformed into specific rules as part of the policies. In addition, the language of the guidelines is suitable for software engineers as it gives them a choice of selecting a suitable technique to implement the properties, e.g. the choice of trust model.

The discussion with experts in Section 5.4.3 showed that trust is a vague property that experts were not sure how to implement. To address their concern, this chapter

demonstrated the flexibility of DIS in adopting any trust model that captures the needed trust elements. In this case, the model adopted the work of Agudo et al. (2010) to represent the consumers' trust in terms of levels.

The process of modelling, in general, helped in exploring the application of formal methods using Event-B to verify the CPT properties. The modelled DIS shows how the integration process, from the consumer side, can be governed by security policies containing the CPT properties. It also shows how the consistency and correctness of the rules derived from the elements of the main properties of CPT were verified. In addition, the model targets the data leakage threats: DL1, DL2, DL3, DL4, DL14, DL18, DL19, DL22, and DL22, see Tables 4.4 to 4.8.

6.6 Summary

SecureDIS provides software engineers with a set of *informal* guidelines written in natural language to mitigate data leakage threats during the early phases of DIS development. In this chapter, a *formal* approach was applied to model a DIS and its security policies and verify the correctness and consistency of the model. The Event-B method was used to formalise the requirements of the specific policy elements that satisfy the CPT properties. These elements were gradually built up throughout the model by utilising Event-B abstraction and refinements.

Modelling DIS security policies is useful to demonstrate how access to data can be controlled by several conditions, as explained in Section 6.3: the allowed role was specified as invariants 5 and 6, the allowed purpose was specified as guard 6 of the *AccessData Event*, and the allowed trust level was specified as guard 8 of the *AccessData Event*. This minimises data exposure due to the incorrect specification of the security policies, such as: unauthorised access (DL1, DL2, DL3 and DL14), non-compliance with security policy (DL18, DL19), and the misuse of data by authorised consumers (DL4, DL22).

The process of modelling the security policies does not only address the research question RQ2, but also provides insights to software engineers as how to mitigate data leakage threats in DIS. In addition, it assess the extent to which the guidelines are implementable and suitable for software engineers. The next step of this study aims to assess SecureDIS in a real data integration context to reach conclusions regarding its applicability.

Chapter 7

Is SecureDIS Applicable? A Case Study

The experts' final recommendations on SecureDIS discussed the need to customise it to a specific context to demonstrate its usefulness, see Section 5.4.4. This chapter presents a case study in which SecureDIS is applied to a real data integration project. The selected project belongs to a healthcare appraisal organisation in the UK. The case study assesses the degree to which SecureDIS is applicable to the project, which aims to address research question RQ3 concerned with SecureDIS applicability.

Using a case study as a research method is appropriate for studying a phenomenon in its real context, see Section 3.2.3. Most case studies in software engineering are project-oriented (Runeson and Höst, 2009), which aligns with the goal of investigating a data integration project in the light of SecureDIS.

The applicability of SecureDIS is assessed in the case study by a methodology containing five stages. The first stage aims to understand the project under investigation through interviews. The second aims, through questionnaires, to find out the number of the guidelines that are actually applied in the project, namely the *initial degree of applicability*. The third stage conducts analyses from different perspectives to assess the *potential degree of applicability* based on the project's requirements. The comparison between the initial and the potential degrees of applicability are then discussed in a focus group with the project team members as part of the fourth stage, to reach the final degree of applicability. In the final stage, the case study findings are discussed.

The chapter discusses the project selected and the research design used to assess the degree of applicability of the SecureDIS to the project. In addition, it discusses the project's security practices and draws appropriate conclusions.

7.1 Setting the Scene

Before tackling the case study, the terms used are first defined, the objective stated, and the selected case described.

7.1.1 Definitions

- *Real Data Integration Project*: the data integration project that was running and is under investigation by this study. The project was developed by a healthcare appraisal organisation in the UK.
- *Initial Degree of Applicability*: the number of SecureDIS guidelines actually applied and followed by the project, from the perspective of the developers.
- *Potential Degree of Applicability*: the number of SecureDIS guidelines that could be applicable, based on the project requirements.
- *Final Degree of Applicability*: the number of SecureDIS guidelines that are applied by the project, after the confirmation with the project team members.
- *SecureDIS Qualities*: the general qualities of SecureDIS including: the comprehensiveness of its components and its coverage the CPT properties, its ability to customise it to a context, its usefulness, etc.

7.1.2 Case Study Objective

The case study addresses research question RQ3, which proposes assessing the applicability of SecureDIS in a real data integration project. Investigating the applicability can be linked to secure software development and data leakage threats occurring in the project, to gain more insight about data integration in practice. Hence, the objective is essentially explanatory (Yin, 1984; Stake, 1995; Runeson and Höst, 2009), where the application of SecureDIS is to a real-life context. To achieve this objective, a suitable data integration project was required.

7.1.3 Case Selection

An extensive search was conducted to find a suitable candidate for this study that satisfies the scope of SecureDIS. This was not an easy task as companies were reluctant to participate, knowing that the security of their systems would be investigated. However, one organisation eventually agreed to take part.

The case selected is a data integration project that belongs to a healthcare appraisal organisation in the UK. The project integrates several data sources into a repository to

serve different consumers, in which the sources originate from multiple data providers. Data sources, data providers, and data consumers are all outside the organisation's physical boundary. The integration occurs in a 3-tier architecture. In addition, the data collected and integrated belong to the healthcare domain, so confidentiality and privacy concerns are common among the data sources.

To investigate the applicability of SecureDIS to the selected project (as discussed in Section 5.4.4), the following are specified: 1) the specific domain, which is healthcare; 2) the entity whose data needs to be protected, which is individuals' personal data; and 3) the environment of the system, which is a distributed web environment.

SecureDIS was designed for systems with several data providers from different organisations that would have different security, privacy, and trust requirements. However, to find such an entity with such a system, in addition to the entity's willingness to participate in a security-related study, was a difficult task. Therefore, since the selected case satisfies the main elements of the SecureDIS scope, see Section 1.2, the planned objective can be achieved through this study.

7.2 Case Study Design

The following sections explain the design of the case study and the data collection methods used.

7.2.1 The Unit of Analysis

As part of the case study design, it is important to define the unit of the analysis (Yin, 1984), i.e. the case that determines the entity being investigated. Thus, the unit of analysis is the DIS, designed and implemented as part of a project. Investigating the project includes understanding the security, privacy, and trust related activities with respect to the Software Development Lifecycle (SDLC) used to build the DIS resilient to data leakage threats.

It is also practical to define the study propositions (Yin, 1984), if they are present, to know the scope of the case study and ensure its feasibility (Baxter and Susan, 2008). The main proposition of this study is:

Investigating SecureDIS framework and guidelines in a real data integration project demonstrates SecureDIS's applicability.

7.2.2 Participants

To follow the case study design, it is important to define the subject, i.e. participants, of the study that will provide the information (Runeson and Höst, 2009; Yin, 1984). In this case, the participants are the members of the management and the development team working on the data integration project, i.e. the unit of analysis in this study.

The criterion for including participants is that they are working on the project as a manager, developer, system analyst, or consultant. If the participant was not part of the project, they were excluded, even if they were part of the unit developing the project. Four participants were recruited for this study. Since the team was quite small, the participants worked together in overlapping roles and shared their expertise in building the system. Table 7.1 shows the participants by their given IDs (anonymised) and their job descriptions.

Table 7.1: Case Study Participants

ID	Job Description
P1	Manager of the Software Development Unit (SDU)
P2	System analyst and security designer of the project
P3	Developer and user experience designer
P4	Developer

7.2.3 Case Study Methodology and Procedure

Before conducting the study, the organisation's approval was obtained, see Appendix D.1.

The study was planned with five stages, each benefiting from the previous stage's outcomes, the study's conclusions being built gradually. Figure 7.1 shows the stages and the links between them. An overview of each stage's purposes with its outcomes are discussed below.

First Stage — Exploring: This provided an understanding of the data integration project in order to investigate the applicability of SecureDIS. This was achieved by gathering information about the nature of the project. *Interviews* were conducted with participants, and the outcomes of this stage were:

- The nature of the project under investigation, and its architecture represented as a Data Flow Diagram (DFD).
- The organisation's security, privacy, and trust practices.
- The application of security practices to the SDLC.

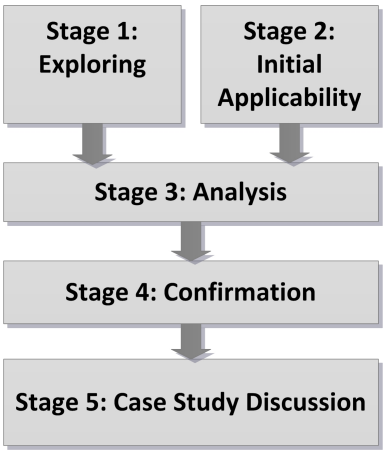

Figure 7.1: The Case Study's Methodology

Second Stage — Initial Applicability: This assessed the degree to which the project complied with the SecureDIS guidelines from the perspective of the developers (a subset of the participants). This was achieved using an online *questionnaire*.

The outcome of this stage was the number of the currently applied guidelines, i.e. *initial degree of applicability.*

Third Stage — Analysis: This analysed the outcomes of the previous stages, including assessing *the potential degree of applicability* and comparing the results with *the initial degree of applicability.* The analysis was conducted from four different perspectives: the SecureDIS components, the Confidentiality, Privacy, and Trust (CPT) properties, the SDLC activities, and the data leakage threats.

The outcomes of this stage were the *potential degree of applicability* alongside an understanding the organisation's security practices. Inconsistencies and ambiguities in participants' responses were also identified for further investigation.

Fourth Stage — Confirmation: This discussed the findings of the third stage with the participants through a *focus group.*

The outcomes of this stage were: 1) reporting the previous stages' findings as a form of a validation (i.e. members' checking (Seaman, 1999)); 2) seeking clarification of the project's nature and security practices not discussed during the first two stages in order to resolve contradictory and neutral responses and so reach a *final degree of applicability*; 3) obtaining feedback of the SecureDIS qualities, such as comprehensiveness and usefulness; and 4) an understanding of the organisation's security practices that helped us provide security recommendations to the team.

Fifth Stage — Case Study Discussion: The final stage discussed the case study findings.

7.2.4 Case Study Management

To ensure the case study satisfied the purpose for which it was designed, several managing elements were defined. The following subsections explain each element.

7.2.4.1 The Study Schedule

The study took nine months, starting from initial approval by the organisation in January 2015 and ending in September 2015. Each activity that required direct engagement with the team members, i.e. organisation time, is explained in Table 7.2. The purpose was to ensure the study was conducted in a way that limited the organisation's time.

Table 7.2: Case Study's Planned Organisation Time

Stage	Activity	Organisation Time
Stage 1 Exploring	Interviews	20-40 min with each member of the team
Stage 2 Initial Applicability	Questionnaires	20-40 min for each member of the team
Stage 4 Confirmation	Focus group	90-120 min with all team members together

7.2.4.2 Ethical Issues

Since the data collected contained information sensitive to the organisation and included private processes, data protection was important. Ethical Approval from the University of Southampton was obtained to conduct the study, with ethics registration: ERGO/FPSE/13819. All voice-recorded data was transcribed and then deleted. In addition, the data collected was anonymised and saved on a password-protected computer. The organisation was given a consent form signed by the researcher to anonymise data and review the results with the team members. All publications resulting from this study required approval from the organisation to ensure data protection.

7.2.5 Data Collection Methods

The data used in the case study was collected using three different methods following a triangulation approach, as discussed in Section 3.2.3. The methods and the rationale behind using them is as follows.

- Interviews: used to explore the views of the team members (Gill et al., 2008) regarding the project.

- Questionnaires: used to collect the views of a sample. Questionnaires designed in a well-structured way allow collecting the same information from participants (ching Leung, 2001).
- Focus groups: used to clarify and extend the data collected, in addition to providing feedback to the participants (Gill et al., 2008).

The following sections explain the details of designing and conducting each data collection method.

7.2.5.1 Interview Design

To achieve the purposes of the first stage, see Section 7.2.3, interviews were employed.

Participants: The project team members listed in Table 7.1 all participated in the interviews.

Material: The interview questions were designed in a semi-structured fashion (Seaman, 1999), with a combination of open-ended and closed questions as follows:

1. **Regarding Data Integration**: Assuming that your current data integration project uses multiple data sources, are the data providers (Tick all that apply):
   - Within the organisation,
   - Outside the organisation,
   - From the same domain,
   - From multiple domains,
   - Other? Explain.
2. Do you use any external entities in the integration process, e.g. cloud services or third party entities (companies or people)? [yes/no] Explain.
3. **Regarding Security**: Are there multiple security, privacy, and trust models[1] used within the project (e.g. security models of data sources or security models of the cloud services)? [yes/no] Explain.
4. Which of the following security standards/guidelines do you follow? (Tick all that apply).
   - None,
   - ISO,
   - NIST,
   - Industry's best practices,
   - Your organisation's guidelines,
   - Other? Explain.

[1]Security policies are a set of requirements that can be represented by a security model (Goguen and Meseguer, 1982)

5. Do you have your own defined trust model? [yes/no] Explain.
6. **Regarding Secure Development**: Do you have a security analysis team? [yes /no].
7. If yes, what process do they follow?
8. Is security analysis combined with the usual tasks of software development? [yes/no] Explain.
9. Do you have a quality assurance team/person that checks the security of the product? [yes/no].
10. If not, how is the security of the product is evaluated?

To ensure the research question RQ3 is addressed properly, three aspects are investigated. Table 7.3 shows the aspects linked to the interview questions.

Table 7.3: Investigated Aspects Linked to Interview Questions

Aspect	Question
The data integration project nature	1, 2
Current security practices	3, 4, 5
Applying security to the SDLC	6 to 10

Procedure: The project's secretary was contacted to arrange interviews for the participants. An email was sent with the information sheet and consent forms to all participants. Upon agreement, meetings were arranged with each participant individually.

At the beginning of the interview, the purpose of the activity was discussed with each participant. Participants were given time to understand and enquire about issues they did not find clear. This was planned to take 5-10 minutes. The questions were asked and clarifications of the responses were requested during the discussion. This activity took about 20-30 minutes, and was conducted in March 2015. At the end of each interview session, the participant was invited to take part in the next stage of the study.

7.2.5.2 Questionnaire Design

To achieve the purposes of the second stage, see Section 7.2.3, an online questionnaire was employed.

Participants: It was planned that all the developers within the team were to participate in this stage (i.e. participants P2, P3 and P4 in Table 7.1). However, this was reduced to two as participant P4 left the organisation during the process.

Material: The questionnaire was presented online[2]. It contained the following parts:

- Information to the participants about the purpose of the questionnaire.
- Introduction to the SecureDIS framework, its architecture, and a short description of each of its components, as presented in Chapter 5.
- SecureDIS guidelines (as presented in Tables 5.21 to 5.26) were included with a scale for the participant to assess the *initial degree of applicability*. This was a 5-point Likert scale, with 'Strongly agree', 'Agree', 'Neither agree or disagree', 'Disagree', and 'Strongly disagree', in addition to an option of 'Not applicable' (see Section 3.2.1 for details of the scale).

Procedure: Each participant was invited by an email that contained the online questionnaire link (see Appendix D.2 for the email message). This process took place during May 2015, and was designed to take 20-30 minutes to complete.

7.2.5.3 Focus Group Design

To achieve the purposes of the fourth stage, see Section 7.2.3, a focus group was employed.

Participants: The remaining three team members all participated in the focus group, as by this stage P4 had left the organisation.

Material: The material of the focus group was as follows:

1. A presentation of the third stage findings including summaries and charts.
2. A number of questions regarding the team's responses to the applicability of the guidelines:
   - *Your responses to guidelines 1, 2, and 26 contradict; what are the reasons?*
   - *Your responses to guidelines 4 and 5 were neutral; what are the reasons? Is it possible that the guidelines are inapplicable?*
3. A discussion of the SecureDIS guidelines applied within the project vs. the potentially applicable guidelines, that would confirm the findings.
4. A set of open-ended questions to understand the team's security practices regarding: purposes of the system, users of the system, data leakage concerns, relying on a permission system, and security policies.

[2]The questionnaire was designed and implemented using the University of Southampton iSurvey service: https://www.isurvey.soton.ac.uk/

5. A discussion of the recommendations to improve the organisation's security practices.

6. A short questionnaire to assess the qualities of SecureDIS, described in Appendix D.5.

Procedure: The focus group meeting was set up in the organisation's work place in mid-September 2015. The meeting was planned to last 90-120 minutes. After notifying the participants of the recording of the meeting, each one was handed the SecureDIS guidelines and the survey to assess the qualities of SecureDIS. The findings of the previous stages of the study were first presented. Participants were encouraged to discuss these findings and ask questions. During the discussion, notes were taken and feedback was given to the team members.

7.2.6 Validity Procedures

To control bias and ensure the study was valid, several steps were taken. First, use of the triangulation technique (Golafshani, 2003; Runeson and Höst, 2009; Mathison, 1988) allowed data to be collected from the different methods to reach a consensus judgement. Secondly, the participants' feedback about the results (Runeson and Höst, 2009) was obtained (called member checks (Seaman, 1999)). These checks were conducted formally via reporting on each response to the applicability questionnaire (2^{nd} Stage), and informally through revising the transcribed interviews (1^{st} Stage). Lastly, a detailed design of the case study (Runeson and Höst, 2009) was circulated that described the steps needed to conduct the study and the reasons behind each step.

7.2.7 Data Analysis Approaches

The case study used a mixed methods approach, both qualitative and quantitative, for each of which the data analysis differs. For the qualitative approach, thematic analysis (Boyatzis, 1998) was employed using NVivo software.

The quantitative approach was used for the questionnaires (both the initial applicability in the 2^{nd} Stage, and SecureDIS qualities in the 4^{th} Stage), with a category scale. The analysis of the questionnaires was not provided for statistical significance, but it was used as a tool for interpreting the qualitative findings.

There are different perspectives on an acceptable sample size in qualitative research (Marshall et al., 2013). In this case, the nature of the study used the project as the unit of analysis rather than people. However, all the project team members who had direct interaction with the project in any form, were included in the sample. Hence, the total was four people at the beginning of the study that became three by its end.

7.2.8 Challenges and Limitations

Several challenges may have had an impact on the way the study was conducted. The first was the difficulty in finding an organisation willing to participate in general, let alone one that employed data integration of confidential data. Organisations are sensitive of their security exposure, and hence were reluctant to participate. The selected organisation was found through an academic setting, and a confidentiality agreement with the organisation was signed including handling their data in an anonymised fashion.

Another challenge was the change from four participants to three (two plus the manager). To mitigate this, the feedback was discussed again with the team members using members' checks to ensure the findings were built on correct facts.

7.3 Results and Findings

Since several data collection methods were used in different stages of the study, the results are presented by stage in the following sections. The participants' feedback is linked to each participant ID shown in Table 7.1.

7.3.1 First Stage: Exploring

The qualitative data from the interview questions were transcribed and manually themed according to Table 7.3. Other aspects discussed were also themed following the data-driven thematic analysis (Boyatzis, 1998). The following headings are the resulting themes, grouped into project nature, current security practices, and applying security to the SDLC.

7.3.1.1 Project Nature

The participants' feedback describing the nature of the project (the interview questions Q1 and Q2) confirms that the business process and the functionality of the system are usually defined and administered by a customer-facing internal unit, called the Appraisal Unit (P3). The Software Development Unit (SDU) (where the integration project is created), is responsible for implementing the system, modelling the business process, and supporting and maintaining the software (P1). Figure 7.2 summarises the tasks of each unit.

External Entities: The project did not require any data integration support from third-party entities and no cloud services were used (P3, P4) to handle the data. However, there were two external entities to the project: 1) the web server hosting

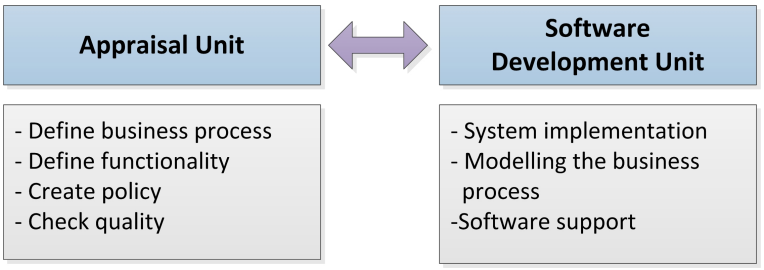

Figure 7.2: Tasks of the Units Involved in the Project

company, and 2) an authorisation entity, namely the General Medical Council (GMC), that validates parts of the data.

Data Sources: The data gathered by the system are described as follows.

- Textual data entered by users using web forms (P1, P3, P4).
- Six evidence documents uploaded by health professionals to prove their good medical practice (P1, P2, P3, P4). The evidence are as follows[3]:
  1. Continuing Professional Development (CPD) document (P1).
  2. Quality improvement activity (Website).
  3. Significant events (Website).
  4. Feedback from colleagues (P1).
  5. Feedback from patients (P1).
  6. Review of complaints and compliments (Website).
- Health professionals' revalidation information provided by the GMC (P2, P3, P4).
- Medical Performers List (MPL) (P2).

The data sources originate from entities/users outside the organisation boundary (P1); however, the data is still within the same medical domain (P2). The data providers and consumers are the appraisers, health professionals, and Regional Officers (RO). Users have login credentials based on their roles.

The Integration Approach: The integration of data sources started with collecting the data and populating a single local database (P3, P4) that resides on the web server. The files uploaded by the health professionals are linked to the database using references to the actual files stored in a repository local to the web server (P4). The local database also includes the login information for each user of the system (P4).

The discussion of the system architecture with the team members led to a DFD explaining how the entities interact, see Figure 7.3.

[3]The data sources are cross referenced with the organisation's website to complete the missing parts of the information. The website is not mentioned to ensure data protection.

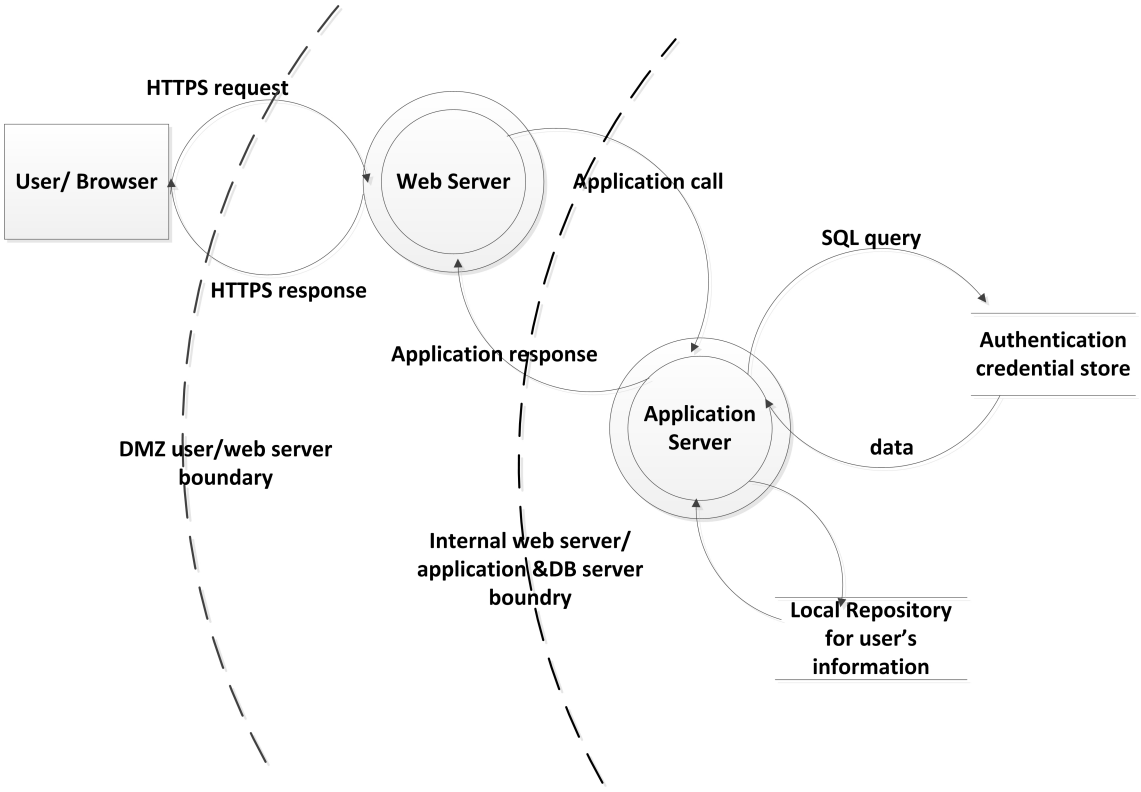

Figure 7.3: DFD of the Appraisal System

7.3.1.2 Current Security Practices

Regarding the organisation's current security practices (interview questions Q3, Q4, and Q5), participants' responses are summarised as follows:

Security Models: To understand the system's security needs, participants were asked about the existence of multiple security, privacy, and trust models within the system. Two opinions were voiced: the majority thought there was only one unified model (P1, P2, P4), while one participant thought there was more than one (P3).

Participants P1 and P2 thought that one level of confidentiality and privacy was required, because health professionals, who enter the system to upload their evidence documents, shared the same level of confidentiality and privacy needs since they belong to the same medical domain. It is the health professionals' responsibility to provide their information to the system and they know the consequences of invalid evidence (P1). Participant P4 explained that the system uses a role-based permission system, each user accessing the information permitted.

In contrast, participant P3 thought that there were several security models in the system: the *revalidation information* data source is provided using a secure SOAP

API[4] to access the data, which participant P2 agreed with. This security model is different from the one designed for the health professionals that requires verifying their information, i.e. being a health professional and belonging to the right region, through the authorising and trusted entity, i.e. the GMC.

Security Policy: A "*blanket policy*", as described by participant P2, that governs the whole system was created for the organisation, namely the Data Management Strategy and Information Security Framework. One of the elements of this policy is to verify the health professionals' identities and the region they belong to in order to be approved and granted access (P3). This is achieved by comparing the unique GMC number provided by the health professional during registration against the number found in the MPL (P2). Participant P3 explained that a personal data policy is not written; however, the security policies and procedures of the project are provided by the Appraisal Unit that governs the project, as mentioned earlier. Therefore, the SDU, where the project is implemented, ensures that the Appraisal Unit's requirements are followed and the SDU is not involved in designing security policies.

Security Guidance: In addition to following the Appraisal Unit's recommendations on the security requirements of the project, the team members have developed their own security practices, described as follows:

- Are adapted from the industry's best practices (P2, P3) and the practices of similar projects in the UK (P1).
- Are adapted from ISO standard (ISO, 2014) but do not completely comply with it, since ISO is resource intensive (P1).
- They do not follow standardised security practices (P2).
- Are adapted from Code Complete (McConnell, 2004), a book containing a secure coding checklist (P1, P2).
- They do not follow any government guidelines (P1).
- Are local to the organisation (P2), i.e. customised to their needs.
- Are not formalised (P4) or fully documented (P3).

On several occasions, participants mentioned the unit's efforts in building an internal quality framework containing security practices that will be used in future projects (P1, P2, and P3). The quality framework contains guidance for the design, development, and implementation phases, in the form of a checklist (P2).

Trust Models: According to participants P2 and P3, no trust model has been created so far, as it does not appear to be needed, especially since there are no external resources (P2) providing the data. However, two entities are external to the project,

[4]API: Application Programming Interface, are a set of protocols and tools used to build web applications.

as discussed in Section 7.3.1.1, and a SLA[5] was used with them. For the GMC, the SLA includes data protection practices, such as encryption requirements, IP address restrictions, etc. (P2). Participant P1 indicated the existence of a SLA with the web server hosting company.

7.3.1.3 Applying Security to the SDLC

To understand how security was incorporated within the SDLC, it was necessary to know more about the team members and their roles in implementing security (through interview questions Q6 to Q10). Three of the team members worked together in designing and implementing the project and reported back to the head of the team, following the Responsible, Accountable, Consulted, Informed (RACI) model of the ITIL[6] (P2). Although participant P2 seemed to be the expert in security (P1), there was no specific member/group responsible for security analysis (P1, P3, P4). All members implemented and checked security as part of their usual development roles.

The DIS was built following an agile waterfall model, and the design and coding phases were carried out iteratively (P1, P2). To find out how security was incorporated within the SDLC, team members were asked to explain the process of applying security measures to each phase. The following summarises their responses.

Analysis Phase Participant P2 said "*we [do] not consider security as part of the documentation in the analysis phase; it is the default to do security as part of the application*", which participant P3 agreed with. However, it is understood that the whole development process must comply with data protection, regardless of the fact it was not documented (P2). Participant P2 explained that the analysis phase did not include any threat analysis. Nonetheless, security threats were addressed during the design and implementation phases.

Design Phase Participant P3 thought that developers think of data protection by limiting data exposure to what is really needed during the design phase; since they know who their customers are, the data should be secure and limited to them.

Implementation Phase The team discussed common security threats and well-known attacks so that they could implement against them (P2, P3). These threats were collected from common knowledge and understanding of security (P2). Participant P3 said: "*we anticipate certain threats, such as SQL injections and session hijacking, so we try and build against those; so we've got certain things in place to stop any efforts for those happening. But it's mainly to stop people getting to the data and to stop hijacking accounts fraudulently*".

[5]SLA: Service Level Agreement

[6]ITIL: Information Technology Infrastructure Library, can be found here http://www.itil.org.uk/

In detail, participant P2 explained that security is implemented by the following:

- Creating a user authentication protocol and creating login information (username and password) that are hashed and stored in a local database.
- Creating an authorisation protocol by defining roles and permissions using RBAC.
- Creating an object-oriented 3-tier architecture, where trust boundaries are defined as follows:
  1. First boundary: between the end user and the application, using authentication and RBAC.
  2. Second boundary: between the application and the database, using authentication.
  3. Third boundary: at the web server level, where the server is protected by authentication.
- Protecting communication by:
  1. Encrypting the communication between the end user and the system using a 128-bit key.
  2. Using security certificates, i.e. SSL, to prevent spoofing, clickjacking, and dictionary attacks.
  3. Lock users upon certain failed attempts of authentication to prevent dictionary attacks.

Participant P2 said that the first step is to build a security protocol of the system as a structure, then to build the functional requirements that adhere to that structure (P4 also agreed to that). Participant P3 added that security is built using session tokens, user checks and validation, and cookies in addition to using the MVC[7] model. The actual coding of the software follows an internal quality framework (P4).

Participant P4 explained how security concerns in full are covered: "*we've not concentrated too much on security of the system at the moment. Permissions are something we are going do retrospectively, so we've implemented basic permissions where you can read a page or not, or write a page or not, but we've left it at that. For the actual permissions on files we have not done that yet. There is a lot we are doing later on*". With regards to downtime, participant P3 said "*we haven't really put anything for that in place with some mechanism to keep our system alive*".

Testing Phase To test the security of the final product, several thorough test cases that include different users' roles were conducted manually (P4). An automated penetration-testing tool was used to check against common security threats, namely

[7]MVC: Model View Controller, is a software design pattern.

Acunetix[8](P1, P2, P3, P4). The tool was also used in different stages of development (P2). The product also went through a quality review conducted by the Appraisal Unit to check the quality of the business process (P1), but not the quality of the technical implementation (P2). Finally, the internal quality framework was used as a reference to check the security of the product (P2), in addition to its use in the other development phases (P1).

7.3.1.4 First Stage Conclusions

The completion of the exploration stage confirmed the initial assumptions about the case's suitability for this study, which was evident during discussion of the project's nature and security practices. The project's requirements and the responses to the interview questions remain within the scope of SecureDIS.

While transcribing the interviews responses, more information was needed to understand the project, hence some parts of the information were cross-referenced with the organisation's website to obtain a fuller picture. More information about internal security practices was sought by requesting the internal quality framework from the project members. As a result, the preliminary findings continue to support the proposition stated in Section 7.2.1.

7.3.2 Second Stage: Initial Applicability

This aimed at finding the *initial degree of applicability* of SecureDIS guidelines to the project, from the perspective of the developers. The impact of the qualitative responses to the questionnaire will be discussed later.

7.3.2.1 Results of the Initial Degree of Applicability

The two developers of the project are the only participants in this stage. Their responses to the applicability questionnaire are summarised in Table 7.4, see Appendix D.3 for the actual responses.

In Table 7.4, 'total' means that both participants agreed or disagreed with a guideline, while 'partial' means that one participant's response differs from the other's. The responses demonstrate that half of the project's security practices have total applicability to the SecureDIS guidelines (20 of the 41 guidelines). In 11 of the guidelines, participants agreed partially to their applicability. The overall result is that the project applies 31 out of the 41 guidelines, or 75.6%.

[8]The tool can be found here https://www.acunetix.com/.

Table 7.4: Responses to the Applicability Questionnaires

No of Guidelines	Team's Responses
31	Applicable to SecureDIS (20 total applicability + 11 partial applicability)
5	Partial disagreement on applicability to SecureDIS
0	Disagreement on the applicability
2	Neutral
3	Contradictory
0	Not Applicable

However, in 5 of the 41 guidelines, the responses indicate a partial disagreement (guidelines 3, 6, 7, 8, and 27), which will require more investigation to confirm the results. In addition, there were no definite disagreements on the applicability of any of the guidelines.

The responses did not include any 'N/A' (not applicable) feedback to any of the guidelines, which can be an indicator that either SecureDIS guidelines were mostly applicable to the project, or the participants did not choose it for other reasons. Therefore, this requires further investigation in Stage 4.

The participants rated guidelines 4 and 5 with a neutral response. This needs further investigation to determine whether the guidelines were inapplicable to the nature of the project, or they were applicable and the project did not employ them for a reason. Because of some contradictions in the responses to guidelines 1, 2, and 26, clarification is needed to explain this.

7.3.2.2 Second Stage Conclusions

By the completion of this stage, the project showed a 75.6% *initial degree of applicability* to the SecureDIS guidelines. Combining these results with the previous stage's findings demonstrates more assurance of the selected case's suitability, i.e. the case study continues to support the proposition stated in Section 7.2.1.

The next stage combined the first and second stages to address the contradictions and neutral responses summarised in Table 7.4.

7.3.3 Third Stage: Analysis

This stage is a qualitative analysis of the findings of the first two stages of the case study. The process of investigating the applicability is as follows:

Step 1: Code the themes from the transcribed interviews in the 1st stage (Section 7.3.1), as well as the SecureDIS guidelines (Section 5.6). Then, link the themes with the SecureDIS guidelines, to find out how they are related.

Step 2: Assess *the potential applicability* of the SecureDIS guidelines to the project, based on the project's requirements.

Step 3: Compare *the potential applicability* with the results of *the initial degree of applicability*, in the 2nd stage, to determine inconsistencies.

Step 4: Conduct an analysis from different perspectives to find out where the focus and the deficiencies of the security practices of the project are. The perspectives are: the SecureDIS components, the CPT properties, the SDLC and security activities, and the data leakage threats.

Each step is explained in the following sections.

7.3.3.1 Step 1: Coding and Linking the Findings

NVivo was used to assist in coding both the themes and the SecureDIS guidelines, see Section 3.2.2. A text search was run to create an initial list of codes based on the investigation aspects shown in Table 7.3. A careful reading of the text resulted in more codes being added to the list giving a total of 28 codes covering the findings of the 1st stage and the nature of SecureDIS and subsequently linked to the SecureDIS guidelines. This list was then divided into four categories, depending on the overall themes (see the tables in Appendix D.4). This step enabled us to draw conclusions and understand participants' responses.

7.3.3.2 Step 2: Assessing Potential Applicability

The potential applicability is based on the project requirements. The following facts about the data integration project are summarised from the 1st stage, see Section 7.3.1.1.

- The data providers are mainly the data consumers.
- No external entity provides data of any sort.
- No external entity is involved in data processing.
- One external entity, the GMC, provides identity verification only, and a SLA is used with it.
- One external entity hosts the web application and a SLA is used to govern the process.

- Data integration occurs in the web server, where the web application is hosted.
- The approach integrates the data in a single relational database.
- Anonymity is not a requirement, as personal data need to be exposed to some users of the system (the appraisers) to verify the identity of other users (the appraisees).

A careful review of SecureDIS guidelines showed that the majority of them (30 of 41) are potentially applicable as they comply with the project's nature and requirements, see Table 7.5. Hence, the *potential degree of applicability* is 73.2%. However, 8 guidelines were inapplicable to the project as they violated its requirements, while 3 guidelines were labelled as 'not sure', as they could be applied if additional requirements are needed as follows:

- When providing feedback to the data provider, guideline 8 would be applicable if the system is concerned with the use of personal data, otherwise it would not be applicable.
- When focusing on adding more privacy techniques, guideline 24 (selecting a privacy preserving technique) and guideline 27 (computing privacy loss) would be applicable if stringent privacy is required, otherwise they would not be applicable.

Table 7.5: Assessing the Potential Applicability of the Guidelines

Category	No of Guidelines	Guideline Numbers
Applicable	30	1, 2, 5, 9, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 25, 26, 28, 29, 30, 31, 34, 35, 36, 37, 38, 39, 40, 41
Inapplicable	8	3, 4, 6, 7, 10, 11, 32, 33
Not sure	3	8, 24, 27

Table 7.6 shows the reasons for the inapplicability of 8 of the guidelines.

Table 7.6: Possible Reasons for Guideline Inapplicability

Reason for Inapplicability	Guideline Numbers
Violating the nature of data sources used in the system	7, 32
Violating the identity exposure requirement	3, 4, 6
Violating the external entities nature	10, 11, 33

7.3.3.3 Step 3: Comparing the Applicability

This step compared *the potential applicability*, from Step 2, with the actual responses of the participants to the applicability questionnaire in the 2nd Stage, i.e. *the initial applicability*, to highlight inconsistencies.

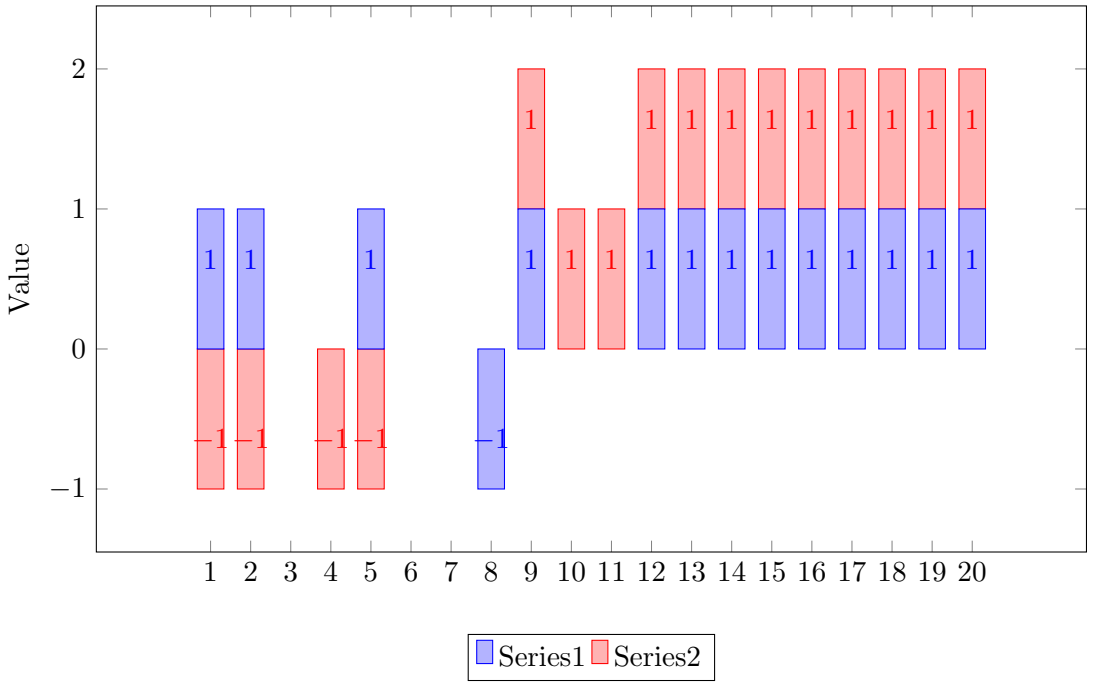

Figure 7.4: Potentially Applicable vs. Initially Applied Guidelines

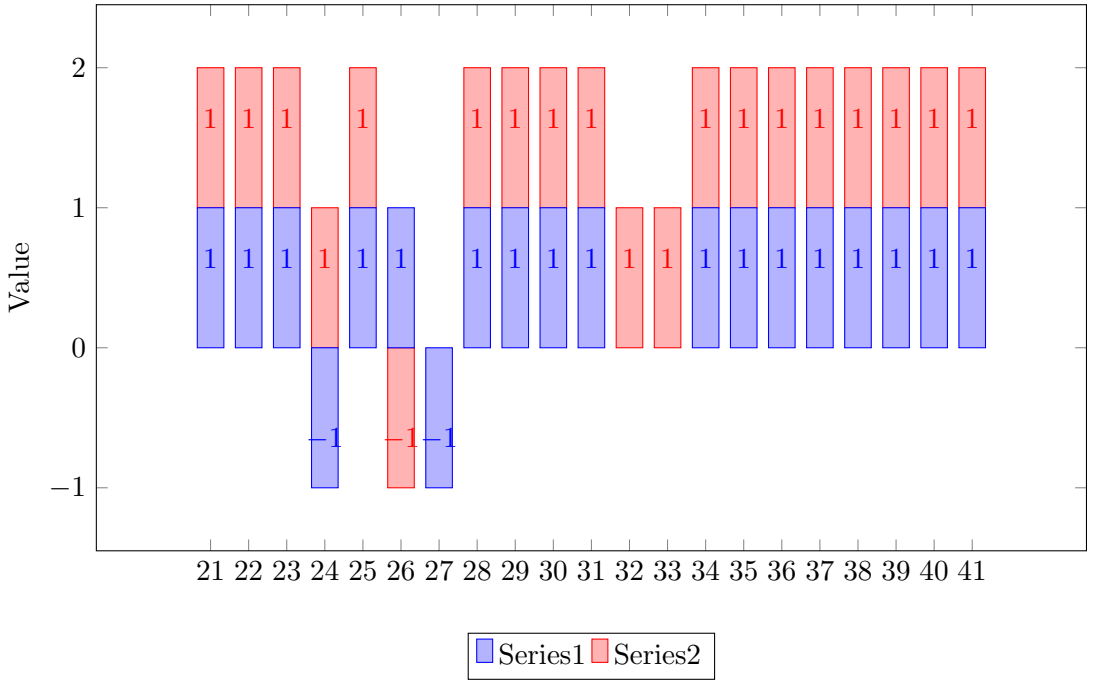

Figure 7.5: Potentially Applicable vs. Initially Applied Guidelines -Continued

To emphasise the inconsistencies, Figures 7.4 and 7.5 are used. The x-axis represents the 41 SecureDIS guidelines, Series 1 shows the initially applied guidelines from the the 2nd Stage, and Series 2 shows the potentially applicable guidelines from step 2. For any guideline:

- If the value is positive, the guideline is either applied or applicable.

- If the value is negative, the guideline's applicability is undetermined and requires further investigation.
- If the value is zero, the guideline is either not applied or not applicable.

Based on the Figures 7.4 and 7.5, further investigation in the next stage is required for guidelines 1, 2, 4, 5, 8, 10, 11, 24, 26, 27, 32 and 33 where inconsistencies were detected.

7.3.3.4 Step 4: Analysing the Findings

This section qualitatively analysed from different perspectives the combination of the quantitative responses to *the initial degree of applicability* questionnaire in the 2nd Stage with the findings of the *potentially applicable* guidelines. The codes defined in Step 1 were used for each analysis perspective to understand the responses and highlight any inconsistencies.

The perspectives used to tackle this are as follows:

1- The SecureDIS Components: This analysis evaluated the applicability of SecureDIS based on its components. Figure 7.6 was produced, where the x-axis represents the SecureDIS components: data and data sources (C1), security policies (C2), the integration approach (C3), the integration location (C4), data consumers (C5), SSM (C6), and external entities (E). Series 1 represents the initial degree of applicability, while Series 2 represents the maximum number of guidelines for each component. This identifies which components lack applicability to the guidelines. The following discussion explains each component in detail.

1) Data and Data Sources: The feedback from the interviews in the 1st Stage discussed both the types of data and the data providers, where the majority of data sources are provided by the users of the system.

In SecureDIS, there are 12 guidelines in the data and data sources category (8 for general data sources nature, and 6 for the data providers, with 2 guidelines overlapping between the two parts). The participants' responses to the applicability of these guidelines were: agreement to 4 of the 12, disagreement with 4, undetermined to 2, and contradictory responses to 2.

The participants disagreed on: guidelines 3 and 6 that focus on applying anonymity techniques, guideline 7 that covers the trustworthiness of the data sources, and guideline 8 that proposes providing feedback to the data provider. The undetermined responses were to guidelines 4 and 5 also related to anonymity techniques. This contradicts the potential applicability findings where guideline 4 was inapplicable and guideline 5 was applicable, according to Table 7.5. The participants had

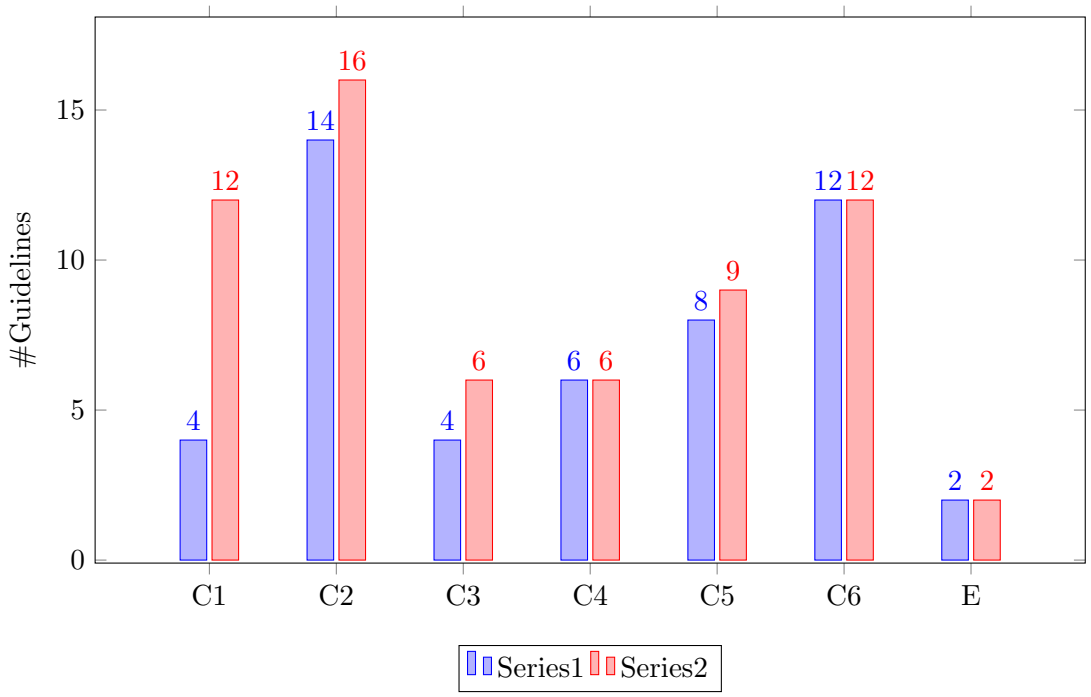

Figure 7.6: SecureDIS Components Perspective

contradictory responses to guidelines 1 and 2 that will require further investigation, which are potentially applicable according to Table 7.5.

Due to the nature of the project, data sources are mostly provided by users and anonymity is not particularly needed. This is because the users need to be identified and their identities need to be verified to be able to provide the correct appraisals for them. Since the users upload their own data, they are held responsible for it, and the trustworthiness measures fall on the users rather on the system.

2) Security Policies: The discussion of this component centred on the security policy and the security models that govern the whole project, see Section 7.3.1.2.

In SecureDIS, 16 guidelines cover the security policies (15 focused on defining policies and 1 focused on having an integrated policy). The participants agreed that 14 of these were applicable to the project. The response was undetermined to guideline 5, which focuses on creating a security policy for each data source. In guideline 26, which focuses on the analysis of query results to predict possible privacy violations, the participants had contradictory responses.

There was agreement with guideline 9 on the need to create a security policy based on integrating all the existing security policies. However, participants had contradictory views on the existence of multiple security models. Their confusion over the number of security models used in the project requires further investigation.

3) The Integration Approach: The views on the integration process was explained by the nature of the project, see Section 7.3.1.1.

In SecureDIS, 6 guidelines fall into the integration approach category. Participants agreed that 4 of these were applicable. Their disagreement with guideline 27 was that there was no privacy-loss computation conducted. A contradictory response to guideline 26 will be investigated later.

The integration approach did not consider the security policies during integration. The privacy-preserving approaches were not mentioned as part of the integration process. However, the agreement with guideline 24 indicates that there are privacy-preserving techniques used within the project based on query analysis, which is contradictory and requires investigation.

4) The Integration Location: The discussions showed that the integration location is local to the web server where that data is uploaded.

In SecureDIS, 6 guidelines focus on this part of the system. The participants agreed to the applicability of all these.

This agreement showed good security practice. However, monitoring and logging were not discussed and can be discussed further with the team members.

5) Data Consumers: The interviews discussed the types of data consumer on several different occasions.

In SecureDIS, there are 9 guidelines in the data consumers category. Participants agreed with 8 of these guidelines, with a contradictory response to guideline 26. While guideline 26 is applicable, based on the potentially applicable analysis in Table 7.5, the contradictory response suggests that it is not clear how the system handles queries. Thus, more investigation is required.

6) System Security Management (SSM): The interviews did not discuss this component. However, the security guidance in Section 7.3.1.2 overlaps with this component, as it provides the guidance on managing and ensuring that the security measures are followed.

SecureDIS contains 12 guidelines focusing on managing security within the DIS. Participants agreed with these entire guidelines as the project complies with them.

There was no discussion of logging and monitoring the integration process within the project, which raises the concern as to whether logging is actually conducted, and whether it checks for the elements suggested in the SSM category. If logging is employed, it would be useful to know the locations of the logs and the analysis mechanism used to search for anomalies. This component will also be discussed later.

The External Entities: In SecureDIS, external entities are the cloud services and the third parties used for accessing, processing, integrating, storing, or managing the data. The participants confirmed that no external entities handle the data in that sense. However, in the project nature description in Section 7.3.1.1, two entities

external to the project were mentioned, one that provides the hosting server, and one that confirms part of the data uploaded by the user.

Only 2 guidelines correspond to the external entities category (guidelines 10 and 11), and there was agreement with both. This was part of the SLA signed with the external entities. This aspect needs clarification with the team members as it violates the potential applicability in Table 7.5 in guidelines 10 and 11.

The conclusion of this analysis is that some of the organisation's security practices require further investigation in the later stages of the case study, such as how the system handles queries, and the mechanism of logging in both the integration location and the SSM.

2- The CPT Properties: In Figure 7.7 the x-axis represents the CPT properties. Series 1 represents the initial applicability of the guidelines based on each property, while Series 2 represents the maximum number possible of SecureDIS guidelines corresponding to each property.

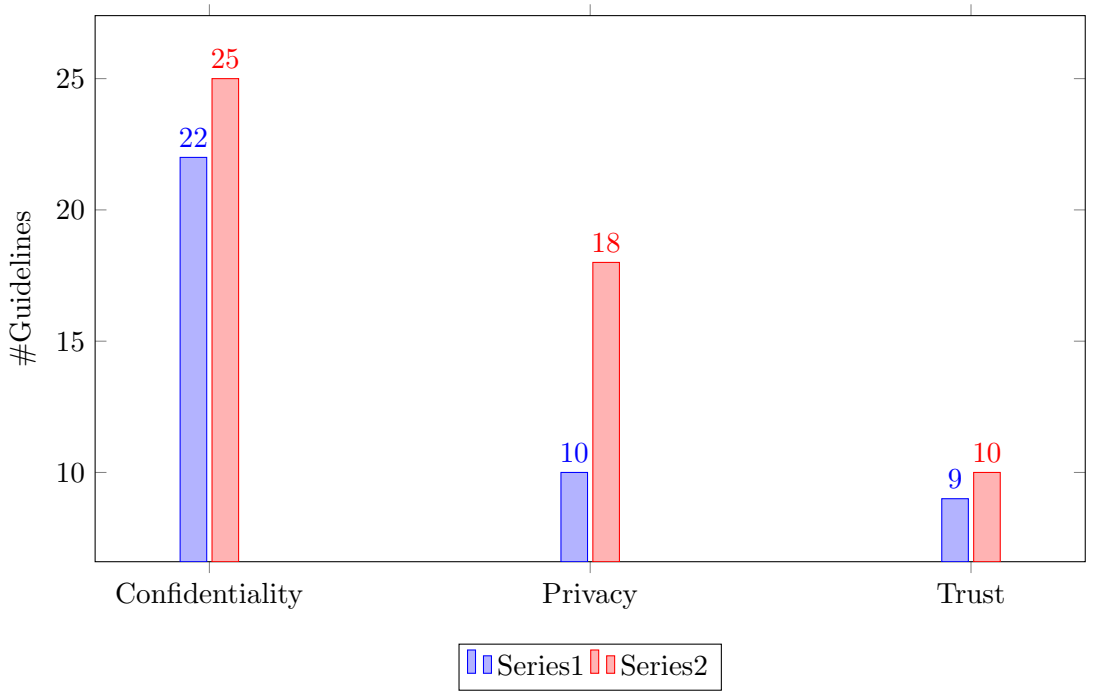

Figure 7.7: CPT Properties Perspective

Each property is discussed by linking the first two stages of the case study together, as follows.

1) Confidentiality: Participants discussed confidentiality as a general concept during their explanation of the security models used in the project, see Section 7.3.1.2. Encryption and access controls, used to implement confidentiality, were covered in the implementation phase of the SDLC, see Section 7.3.1.3. Encryption requirements were also mentioned in the trust models when describing the SLA that governs the external entities, see Section 7.3.1.2.

SecureDIS covered confidentiality in 25 guidelines (14 for generic confidentiality, 2 for encryption, and 9 for access control). Participants agreed that 22 of these were applicable. The response to guideline 5 was undetermined, and participants had contradictory responses to guidelines 1 and 26.

The contradictory responses to guideline 1 indicate the possibility of not having security meta-data defined for each data source. This could be linked to the undetermined response to guideline 5 that suggested creating a security policy based on the security meta-data. This aspect needs to be discussed further with the team members to reach valid conclusions.

The contradictory responses to guideline 26, related to security policies, needs further investigation.

The organisation seems to be implementing confidentiality by focusing on the RBAC model, where implementation is based on a permission system. However, encryption was mentioned in passing as a technique to secure the general communication between users and the system, and to secure the web application's communication with the server via SSL.

2) Privacy: Privacy was not discussed except to state the existence of multiple security models that could include anonymity and data use. However, data use was discussed as part of the data sources, the integration approach, and the security policies, see Sections 7.3.1.1 and 7.3.1.2.

SecureDIS covered privacy in 18 guidelines (12 for generic privacy, 3 for anonymity, and 3 for data use). Participants agreed with 10 of these, disagreed with 4, were undetermined on 2, and were contradictory on 2.

The disagreement with guideline 8 was that the system did not provide feedback to data providers concerning its data use. However, participants agreed to guidelines 14 and 31 that ensure enforcing and defining the data use purposes within the system. In guideline 2, which focused on checking the purpose statement, participants gave a contradictory response. Most of the data provided to this project belongs to the users who uploaded the data to be reviewed for one purpose, the appraisals. Hence, the system does not require this feature. However, if the system was created for several purposes, then this property needs to be implemented to ensure that the provider is aware of why, when, and how much of their data was used.

In terms of applying privacy-preserving approaches, the disagreement with guideline 27 indicates there are no privacy loss computations. This may have a link to the undetermined response to guideline 5 that suggests creating a security policy that should include privacy loss. In terms of anonymity, participants disagreed on using any anonymity technique in guidelines 3 and 6. This can be linked to the undetermined response to guideline 4 that suggests removing QIDs that may expose

the identity of a person. The contradictory response to guideline 26, which focuses on the analysis of query results to predict possible privacy violations, requires further investigation.

Privacy seems to be followed when it is about defining the data use purpose. However, anonymity techniques are not applied. This corresponds to the results of the potentially applicable guidelines for guidelines 3,4, and 6 in Table 7.5.

3) Trust: Within SecureDIS, trust includes the trustworthiness of an entity and the use of trust models. Participants had two different definitions of trust. Participants indicated earlier (see Section 7.3.1.2) that there was no need to define trust models as this project had no external entities that provided data (although there were external entities that handle part of the system). However, when discussing the implementation phase, Section 7.3.1.3, participants expressed trust as being boundaries within the system architecture demonstrating different privilege levels.

SecureDIS covers trust in 10 of its guidelines. Participants agreed to the applicability of 9 of the guidelines and only disagreed on guideline 7. Guideline 7 focuses on monitoring the data sources to determine their trustworthiness levels, but participants indicated that trustworthiness in data sources was not assessed. However, the participants agreed on the applicability of guideline 32, which suggests establishing trust with data providers, which contradicts the results in Table 7.5.

The definition of trust was thus not uniform among the participants; their responses may have different interpretations. However, there was agreement on not defining trust models within the system. The team are using a SLA to ensure the entities accessing the data are bound to protect the data and did not rely on a clearly defined trust model or any technical implementation to check that trust was not actually violated.

Authentication was a security property implemented in the system. Authentication by user names and passwords was employed by users and within the layers of the system, particularly between the application and the database, and within the web server. This property is out of scope of the SecureDIS guidelines.

The analysis based on the CPT properties shows that the organisation's practices in terms of enforcing confidentiality seem to be sufficient. However, there was a lack of defining trust models governing external entities. Due to the requirements of the project, privacy was not a property of interest.

3- The SDLC and Security Activities: Participants indicated that the incorporation of security started with the design phase, which is different from the concept of security by design that propose starting at the analysis phase. The team's responses to the SDLC were grouped according to the development phase in Section 7.3.1.3.

In Figure 7.8, the x-axis represents the SDLC phases. Series 1 shows the number of the guidelines applied in each phase, while Series 2 shows the maximum number of guidelines in each phase.

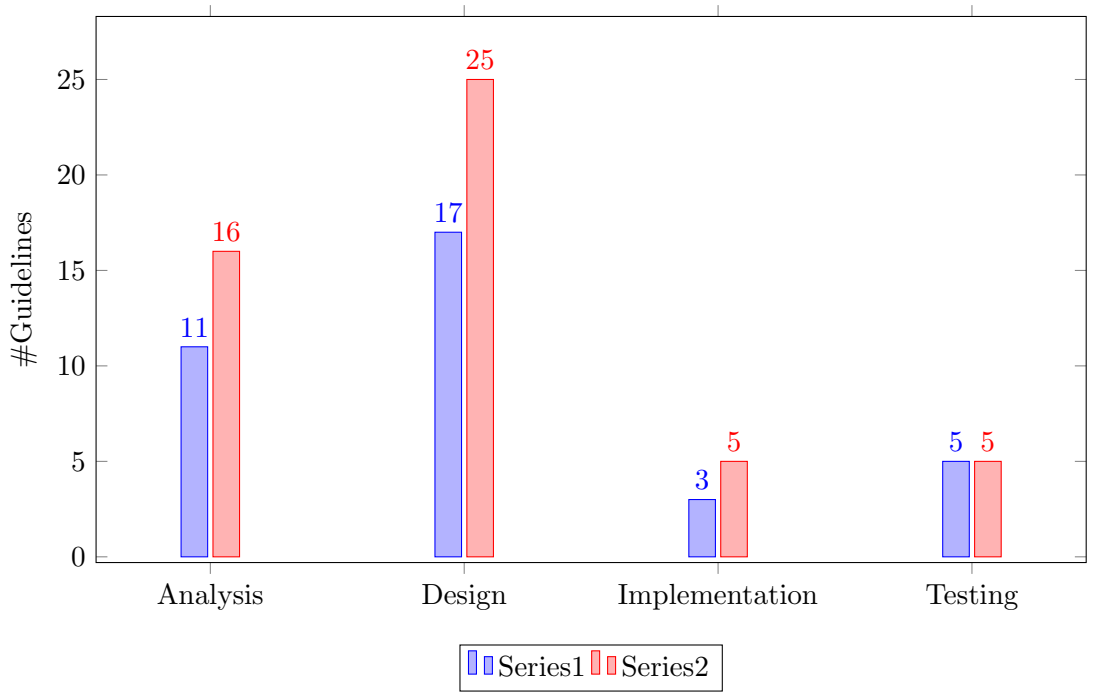

Figure 7.8: SDLC Phases Perspective

The participants' responses to the first two stages are now explained.

1) Analysis Phase: Participants indicated their prior knowledge of data leakage threats but no actual security/threat analysis had been conducted. SecureDIS contains 16 guidelines that cover the analysis phase. Participants agreed to the applicability to 11 of these. They disagreed on 2: guideline 6 focuses on analysing the type of data to prevent inferences attacks, while guideline 7 covers the investigation of the trustworthiness of the data sources. Participants were undetermined on guideline 4, which proposes using an analysis technique to enforce anonymity. Contradictory responses to guidelines 1 and 2 will be investigated later.

2) Design Phase: Participants showed an increased effort to avoid known security threats when designing the system, such as SQL injections and session hijacking. However, the threats addressed in this phase were based on the knowledge and expertise of the team members and not on the results of a threat analysis.

SecureDIS contains 25 guidelines that focus on applying confidentiality and privacy techniques to the DIS in the design phase. Participants agreed to 17, disagreed with 3, had undetermined responses to 2, and contradictory responses to 3, namely guidelines 1, 2, and 26.

The disagreement was on guideline 3 that suggests using an anonymity technique, guideline 8 that suggests providing feedback to the data provider, and guideline 27 that suggests calculating privacy loss, which were all discussed earlier.

3) Implementation Phase: The participants explained their implementation process in detail as well as their internal quality framework that suggests particular security implementation techniques.

SecureDIS contains 5 guidelines that focus on implementing security. The participants agreed with 3, disagreed with guideline 27 (discussed in the design phase), and contradicted guideline 26.

4) Testing Phase: The participants discussed their organisation's approach in testing security measures. The project used Acunetix to conduct web application penetration testing, as well as their internal quality framework.

SecureDIS contains 5 guidelines that focus on testing security measures. Participants agreed on the applicability of all 5 guidelines.

The conclusion of this analysis perspective was that in each phase more than 50% of the guidelines were applied. Hence, no particular phase needed discussion except the analysis phase, where the threat analysis was not conducted.

4- Data Leakage Threats: This perspective determined which guidelines were not applied and how they could impact the security of the system. The guidelines are as follows.

- The potentially applicable guidelines; these were not applied or had neutral or contradictory responses: 1, 2, 5, and 26.
- The potentially inapplicable; these had a neutral or response: guideline 4.
- Those with undetermined responses either in the analysis or in the questionnaires are: guidelines 8 and 27.

The SecureDIS guidelines not applied are cross-referenced in Tables 5.21 to 5.26 to determine the links to the data leakage threats (see Tables 4.4 to 4.8). The results are summarised in Table 7.7.

The confidentiality and privacy in guidelines 1, 2, 5, and 27, corresponding to leakages DL18 and DL19, are possibly violated. The lack of determining the required security and privacy of the data through the use of security meta-data is as important as not capturing the security and privacy requirements. Both vulnerabilities lead to violation of the data source's confidentiality or privacy. This data leakage is mainly caused by ignoring the value of the data and its sensitivity.

Table 7.7: Guidelines Linked to Data Leakage Threats

Guideline No.	Data Leakage Threat ID
1, 2, 5	DL19
4	DL7, DL8, DL9
8	DL24
26	DL7, DL8, DL9
27	DL18

Privacy can be violated in guidelines 4 and 26 corresponding to leakages DL7, DL8, and DL9. The leakage can occur by ignoring the possibility of inference attacks through exposing parts of the data more than defined in the privacy requirements, or by not analysing query results before returning them to the consumers. Privacy can also be violated in guideline 8, corresponding to leakage DL24, when data is used contrary its intended purposes without informing the data provider.

To demonstrate SecureDIS usefulness, the guidelines not applied were discussed with the team members in the next stage to think about the possibility of the data leakages.

7.3.3.5 Third Stage Conclusions

The third stage of the case study focused on the analysis of the previous two stages. The following conclusions were drawn:

- These guidelines require another response from the team members regarding their applicability:
  - guidelines 1, 2, and 26 had contradictory responses.
  - guidelines 4 and 5 had neutral responses.
- The initial degree of applicability (75.6%) differs from the potential degree of applicability (73.2%). This is evident in 12 guidelines: 1, 2, 4, 5, 8, 10, 11, 24, 26, 27, 32, and 33. This difference is addressed in the next stage to reach *the final degree of applicability.*
- Some security practices were not clear, such as handling queries, system logging, etc., that require further investigation.

The organisation's good security practices in considering confidentiality was revealed by the different analysis perspectives. The definition of trust needs to be investigated, as well as the meaning of privacy to the organisation, which would explain why some of the security activities were overlooked.

The next stage of the case study will report these findings to the team members for confirmation and to clarify the responses to the guidelines.

7.3.4 Fourth Stage: Confirmation

This stage employed a focus group of the team members. The conclusions of the previous stage in Section 7.3.3.5 were presented to the team. Several aspects were discussed including: inconsistencies in the responses, the applicability of guidelines compared with those actually applied, the organisation's security practices, the security recommendations to the team, and the reflection on SecureDIS applicability and usefulness. The following subsections report on each aspect.

7.3.4.1 Discussion of the Responses

Participants had provided responses in the 1st and 2nd stages that required further investigation. Some of these responses were contradictory or neutral. Hence, the focus group aimed to clarify the outstanding points.

Participants had no clear response for **guidelines 1 and 5** that rely on each other. Guideline 1 suggests checking for security meta-data for each data source. The meta-data for a source was proposed for use in creating a security policy for that source (guideline 5). The team agreed on the applicability of both guidelines, reasoning their lack of clear response was not calling those requirements meta-data. The team does check for security meta-data but the meta-data is not stored anywhere in the code.

Regarding **guideline 2** that proposes checking for privacy requirements, the team agreed to check for those requirements being bound by the Data Protection Act (UK Parliament, 1998) (P1). Hence, they collect and process personal data for a purpose that is communicated clearly to the data owner. Participant P2 explained that the user agrees to the terms and conditions of data use, and the developers comply with that. The data use is also checked by the Appraisal Unit (P1).

Participants disagreed on the applicability of **guideline 26**, analysing the query results after the integration. Participants P2 and P3 explained that once a user is authenticated, they are automatically authorised to obtain the results (i.e. the summary of the appraisal). Participants P1 and P2 said that, after the results were obtained, the checks carried out were business rules rather than security checks.

Due to the nature of the project, the team disagreed to the applicability to **guideline 4** that suggested removing the QIDs. Participant P1 thought that it was not applicable because "*the data we collect is a record about a particular person. Once we have a unique identifier for them, the system uses it. The whole idea of what we are storing is that people can be personally identified. That is important because the appraisal is linked to a person and the revalidation is linked to a person, so both business functions are linked to the individuals. It's important that we don't have any dissociation*". Participants P2 and P3 thought that some QID analysis is conducted but not across other databases.

7.3.4.2 Discussing Applicable vs. Applied Guidelines

The analysis of Section 7.3.3.3, which compared the guidelines' *initial applicability* and the *potential applicability*, was presented to the focus group. Some guidelines required further investigation as their applicability was marked as 'not sure', see Table 7.5.

- **Guideline 8** that suggests providing feedback to the data provider about the use of personal data was actually applied, contradicting the responses to the 2nd Stage. Participant P1 explained, "*feedback is provided to [the data providers] at the inception of their relationship [with the system], following the Data Protection Act*". The summary of the appraisals is also a form of a feedback to the customers.
- **Guideline 24** suggests selecting a privacy-preserving technique before the integration was applied. All participants agreed that their privacy-preserving techniques are based on the user roles. Participant P3 explained, "*when the health professional sends information, we are preserving the privacy of all other health professionals. The health professional cannot see other health professionals' data*". In addition, the ROs can only see the health professionals that belong to them.
- **Guideline 27** suggests computing the privacy loss was not applied, although it is applicable according to participant P1.

Whilst several guidelines were marked as 'inapplicable' in Table 7.5 due to the nature of project, they were contradictory to the responses of the 2nd Stage, the guidelines were discussed to confirm the final responses as follows:

- **Guidelines 10 and 11** suggest defining third parties' rights on data and establishing trust with those entities, were indicated as applied. Participants' responses to the guidelines were from the perspective of the web hosting company that stores and handles their data. Participant P1 explained that the agreement with the web hosting company strictly states that the company has no rights on the data except for secure disposal in cases of contract termination. P1 continued that the trustworthiness criteria followed by the organisation includes the company's reputation, track record, client base, etc. Therefore, the responses were changed to 'applied' instead of 'inapplicable'.
- **Guideline 32** suggests establishing trust with the data providers, which was applied. The reason behind the response, said P2, was that trust was implemented by validating the health professionals via email and with the external entity via ID number. However, this was not the definition of trust used in the guideline. Therefore, the final response was changed to 'not applied'.

- **Guideline 33** suggests ensuring trustworthiness of the integration services, which was applied. Participants P2 and P3 agreed on not providing the data to any service. Hence, this guideline became 'inapplicable'.

Table 7.8 summarises the responses to the guidelines' applicability before and after this confirmation.

Table 7.8: Guidelines Discussed by the Focus Group

No.	About	Potential Applicability	Initial Applicability Response	Final Response
1	Security meta-data	Applicable	Contradictory	Applied
2	Checking for privacy requirements	Applicable	Contradictory	Applied
4	Removing QID	Inapplicable	Neutral	Inapplicable
5	Creating security policies for data sources	Applicable	Neutral	Applied
8	Providing feedback about the use of personal data	Not sure	Not applied	Applied
10	Defining third parties' rights on data	Inapplicable	Applied	Applied
11	Establishing trust with third parties	Inapplicable	Applied	Applied
24	Selecting a privacy preserving technique before the integration	Not sure	Applied	Applied
26	Analysing query results	Applicable	Contradictory	Not applied
27	Computing privacy loss	Not sure	Not applied	Not applied
32	Establishing trust with data providers	Inapplicable	Applied	Not applied
33	Ensuring trustworthiness of the integration services	Inapplicable	Applied	Inapplicable

7.3.4.3 Understanding Security Practices

The study investigated the project's business process to understand the security and privacy practices employed by the organisation. Participant P1 stated that the main purposes of the system are the appraisals and the revalidation. It also has a future use for data sharing and collaboration with other projects using the collected databases.

Another aspect discussed by the focus group was enquiring about *what the organisation mainly protects*. Participants agreed on protecting the organisation's reputation by

protecting against unauthorised access to personal data and the exposure of the system's business process.

The *users of the system* were discussed. Data providers and consumers are both appraisers and health professionals. Other users of the system are consumers, who do not provide data, including Appraisal Coordinators (AC), ROs, management staff, and administrators.

The organisation's security concerns regarding *data leakage* were addressed. Participants P1 and P2 believed that, although they have not experienced any data leakages, they are concerned about them. Participant P3 was concerned about the users' roles. Participant P1 explained that the technical implementation on the server side, and some of the administrative procedures, may have an effect on the security of the system, both beyond the software development unit's control.

The organisation's reliance on the *permission system* as their main security measure was discussed. The participants thought that its use was complimented by the business rules that maintain privileges and ensure data is only viewed by the right people. The integration process was also governed by these measures. Participant P2 explained that users who query the integrated data need to be authenticated by successfully communicating with the database. Participant P3 pointed out some of the roles have access only to parts of the data.

Finally, the *security policy* of the organisation was addressed. Participant P1 explained that their security policy statement is derived from the organisation's security framework. The statement follows the Data Protection Act and is added to the unit's internal quality framework, as stated by P2. Participant P2 added that the technical aspects of the policy were usually interpreted by the unit, and might require more documentation for audit purposes. Participant P1 said that security policy was addressed in the unit's data management strategies.

7.3.4.4 Security Recommendations

Following the clarification of the organisation's security practices, several recommendations were discussed with the team members to improve their security activities:

- SecureDIS emphasises the idea of security by design. Within the project, security is incorporated into the SDLC starting from the design phase. However, the practice of applying security can be improved by conducting a threat analysis during the analysis phase of the project and prior to its design, to address threats earlier in the cycle.

- During the design phase, the team designed the system bearing in mind the common threats that they have experience with. They are advised to keep up-to-date knowledge of possible threats using reliable sources. It is important they also understand the vulnerabilities in the tools and the infrastructure used to build the system by checking the CVE[9] database or the exploit database archive [10]. It is recommended that they study common attack patterns, so that they can be avoided during design, by reviewing recent online databases, for example the CAPEC[11] database.

- A security analysis needs to be conducted thoroughly to investigate possible confidentiality, privacy, and trust threats and violations that can occur in the project considering all the entities interacting with the system. This is what SecureDIS aims to achieve.

- It is recommended that the team members build secure systems by investigating the software weaknesses and vulnerabilities that allow attacks to occur. This is possible using the CWE[12] database. The team were advised to develop a security library that can be reused in the code, where SQL injection and other common threats are considered.

- The team seems to be reliant on one product for penetration testing. It is suggested that they compare the outcome of that tool with other products, such as Burp Suite[13], to gain more coverage of any possible threats.

- In particular, the project does not really focus on privacy. Hence, developers are advised to look into understanding inference attacks and their implications on data privacy.

- A trust model needs to be defined for entities and components outside the organisation. This is useful so that specific criteria are available to assess the trust level of an entity and avoid data leakage from it.

- The organisation relies on a SLA to ensure the data is kept confidential and private. It is recommended that a technical approach is used to ensure confidentiality. An example would be logging the access to data outside the web server to detect anomalies in accessing the data. Another example would be evaluating the trustworthiness of the company providing the hosting services by monitoring aspects such as behaviour and reputation.

- It is recommended that all system activities be logged in different locations and the logs analysed using the tools available.

[9] CVE: Common Vulnerability and Exposure, https://cve.mitre.org/.
[10] Can be found here, https://www.exploit-db.com/.
[11] CAPEC: Common Attack Pattern Enumerations and Classification, https://capec.mitre.org/.
[12] CWE: Common Weakness Enumeration https://cwe.mitre.org/.
[13] Can be found here, https://portswigger.net/burp/.

- It is recommended that the UK government security essentials[14] be followed, or similar government guidance.
- Selecting a specific security technique, such as SSL, over another requires further investigation and knowledge of its configuration.
- It is important to combine the security policies of the entities involved in the DIS. This is evident when there are multiple entities located to different countries where data protection regulations vary. This also applies when combining data coming from different domains that are subject to different regulations in the same country. In this study, the project required combining data within the same country and within the same domain; hence, the importance of combining the security policies cannot be seen. However, this is our recommendation for future projects.

7.3.4.5 Reflection on SecureDIS

The discussion of SecureDIS with the team members revealed the need for a glossary to accompany the framework. Although the team members were knowledgeable and specialised in software engineering, there was still a need to define the terms used in the framework precisely. Hence, this recommendation will be considered for the future use of SecureDIS.

The participants had responded to a short questionnaire, detailed in Appendix D.5, that contained three parts: *the first* assessed whether SecureDIS covered all the components and all the CPT properties. The results showed that two of the three participants agreed component coverage was comprehensive, while there was general agreement on the coverage of the CPT properties.

The second part assessed the qualities of SecureDIS. With varying levels of agreement, all participants thought that SecureDIS was: 1) practical, 2) implementable, 3) could help in mitigating data leakage threats, 4) and could be used by software engineers. Regarding the ability to apply/customise SecureDIS to a context (i.e. a real project), two of the three participants thought it was possible and one participant was not sure.

The third part assessed the usefulness of SecureDIS. For details of the participants' responses, see Appendix D.5. The results are summarised as follows.

- 1/3 thought it could help them accomplish tasks, while 2/3 were unsure;
- 2/3 thought it could improve their performance, while 1 was unsure;
- 2/3 thought it could increase their productivity, while 1 was unsure;

[14] Cyber Security Essentials Scheme, can be found here: https://www.gov.uk/government/publications/cyber-essentials-scheme-overview.

- 3/3 thought it was effective;
- 3/3 were unsure whether it could make their job easier;
- 2/3 thought it was overall useful.

7.3.4.6 Fourth Stage Conclusions

This stage discussed the findings with a focus group of team members to check inconsistencies in the responses. The number of guidelines applied increased to 32, from 31 in the previous stages, and the ambiguous responses were resolved into 2 inapplicable guidelines and 7 guidelines not applied. The final degree of the applicability of SecureDIS to the project is thus 78%.

The organisation's security practices were discussed and a few recommendations were provided to the team. The team received insight into conducting a proper threat analysis that accommodates the details of the business process to mitigate data leakage threats.

At the end of the case study, a report was sent to the team manager for review before including its contents here. The internal quality framework was also reviewed and security recommendations added to it.

7.3.5 Fifth Stage: Case Study Discussion

The study to assess the SecureDIS applicability to a real context showed that 78% of the guidelines were applicable. This contrasts with the *potential applicability* assessment, where 73.2% of the guidelines were assumed to be applicable based on the project requirements, see Section 7.3.3.2. Some guidelines required further investigation. However, during the initial questionnaire, the team members thought that they applied 75.6% of the guidelines, see Section 7.3.2.1. The reason for these differences was a misinterpretation of the guidelines, arising from differences in terminology used in SecureDIS and by the team members.

Although the final applicability of the guidelines was 78%, the remaining guidelines could be applicable if they were part of the project requirements. Therefore, the applicability degree is closely related to the project requirements, the nature of the context, and the level required from each of the CPT properties. It is not expected from SecureDIS guidelines to be applicable in total to any given scenario, as explained earlier in Section 5.7.

The case study findings were based on several data collection methods that helped gain confirmation of the findings and resolve inconsistencies and ambiguities. The study supports the proposition in Section 7.2.1 by investigating SecureDIS applicability and

therefore answering research question RQ3 that proposed using a real data integration project to conduct the assessment.

Generally, SecureDIS was positively received by the team members, the target audience. This is evident in SecureDIS's usefulness in highlighting the data leakage threats and emphasising the CPT techniques, especially in discussing the need for a trust model and the consequences of interference attacks. In addition, the study aimed to understand the application of SecureDIS through recognising the project security requirements and their link to SecureDIS guidelines. The applicability assessment addressed the experts' concerns discussed in the reviews in Section 5.4.4, which suggested applying/customising SecureDIS to a context to evaluate its applicability.

Although case studies are usually time-consuming and require a lot of contact and management with the people involved, they are suitable research approaches for an explanatory study (Runeson and Höst, 2009). Since SecureDIS has already been developed, a case study can explain how the guidelines can be implemented in reality. However, the use of a case study as a method for applicability assessment has its benefits and drawbacks. The study helped bridge the gap between theory and practice and provided better understanding of how SecureDIS was received by the target audience. The case study also highlighted how security practices were employed in practice and how organisations think about security during development. Also mentioned was the fact that the security policies that came from a different unit were interpreted by developers in several ways.

It was noted that threat analysis was not formally carried out in this particular project, and possibly in many other organisations. This opens up a research direction of investigating the ways in which organisations address threats in practice in the absence of a threat analysis process. One aspect that always needs to be conveyed to developers and policymakers, is understanding the value of data and the consequences of its exposure. This ensures better security consideration during system development.

The study helped highlight the importance of including a glossary with the future use of SecureDIS, arising out of the need to explain the guidelines in detail to the target audience. Providing enough supporting documents with SecureDIS would help users to understand the guidelines properly and would improve SecureDIS's usefulness.

The organisation that was the subject of this investigation also benefitted. The interaction with participants improved their awareness of data leakage and its consequences, especially from the privacy and trust aspects. It also provided them with an opportunity to review their security practices with an outsider, providing a different perspective.

7.4 Summary

This chapter presented a case study to assess the degree of applicability of SecureDIS to a real data integration project. The case study was conducted in five stages: 1) understanding the project nature through interviews, 2) assessing the initial degree of applicability of the project to the SecureDIS guidelines through questionnaires, 3) assessing the potential applicability and comparing it with the initial applicability, and conducting analyses from different perspectives to understand the project practices, 4) confirming the findings with a focus group that led to the final degree of applicability, and 5) discussing the case study findings.

Three different assessments of applicability were conducted within the case study at different stages:

- The *initial degree of applicability*, in Stage 2, was based on the questionnaires, revealing 75.6% of the guidelines were applied.
- The *potential degree of applicability*, in Stage 3, was based on understanding the project requirements, showing 73.2% of the guidelines were potentially applicable.
- The comparison between the previous assessments, in Stage 4, led to the *final degree of applicability* in which 78.0% of the guidelines were applicable to the project.

The applicability results addressed research question RQ3 that aimed to assess the SecureDIS applicability in practice. It also addressed the experts' concerns regarding the ability to apply SecureDIS in practice, see Section 5.4.4.

The case study demonstrated that SecureDIS can be used by software engineers. Although a glossary needs to accompany the framework to explain the terminology, SecureDIS was generally understood by the developers and analysts of the project. In addition, the study demonstrated how SecureDIS is useful in identifying data leakage threats that were overlooked by software engineers.

Overall, the choice of the case selected for the study was suitable. The case covered the scope of the SecureDIS. Also, the degree of SecureDIS applicability to the project is an indicator of the case's suitability.

Chapter 8

Conclusions and Future Work

This chapter summarises the results and the findings reached to answer the research questions. It also discusses the contributions made by this research. The future research directions are also explored.

8.1 Conclusions

DIS that integrate sensitive and personal data from multiple and heterogeneous data sources across organisations are associated with data leakage threats. This is possible through failing to maintain the security and privacy requirements and by not considering the trust levels of the entities involved in the integration process. In addition, these systems are prone to unauthorised access, secondary misuse of the data, violation of personal data protection, and fail to capture the required levels of the data providers' security policies. Therefore, there is a need to acknowledge those kinds of system and to mitigate data leakage threats for the sake of data value and the risks associated with its exposure to unauthorised entities.

The first research question focused on eliciting the types of threats that DIS are prone to, see Section 3.1. The focus was mainly on data leakage threats that violate the data's confidentiality and privacy in addition to trust in system entities, i.e. the CPT properties. However, to be able to conduct the analysis, a conceptualised architecture of DIS with middle layers was needed. A preliminary architecture based on a literature review was therefore created. This was evaluated by experts in the field to ensure its coverage of the main components and to anticipate possible data leakage locations within the architecture. Chapter 4 focused on the process of threat analysis and resulted in identifying 25 threats. The results of the analysis showed that the threats mainly arise from mis-design of the CPT properties of the system and by the lack of capturing and of enforcing the security policy within the integration process.

Associating security threats with flaws in the system design encourages combining the fields of software engineering and information security. Therefore, the aim is to target software engineers who wish to build secure systems from the start by employing the concept of security by design. The second research question aims to find an approach that helps software engineers to build secure DIS by mitigating the data leakage threats elicited earlier, see Section 3.1. This can be achieved by having clear security requirements propagated through software development to produce a DIS that is less vulnerable. The proposed mitigation approach is a novel framework, SecureDIS, as suggested by Clifton et al. (2004). The reason for creating a framework as an approach was to allow software engineers to have more flexibility in choosing the components they wish to focus on, in addition to addressing the experts' suggestion of creating a framework that allows flexibility in choosing techniques.

While the proposed solutions to mitigate data leakage cover very specific components of DIS or a specific security property, SecureDIS considers the security of the main components of DIS with middle layers to ensure that security is in place. It also includes external entities, such as cloud services and third parties. The framework was detailed in Chapter 5, and its components are: 1) data and data sources, 2) security policies, 3) data consumers, 4) integration approach, 5) integration location, 6) System Security Management (SSM). For each component of the SecureDIS there is a set of design guidelines addressing a particular data leakage threat, and its links to the CPT properties. SecureDIS encourages developers to design a secure DIS by the use of CPT techniques and by involving security policies as part of the data integration process. The language of the guidelines was designed to be understood by software engineers and to provide enough detail to allow flexibility in choosing the proper implementation of those guidelines. The guidelines were confirmed by the experts. Discussion with the experts resulted in the requirement to apply the proposed guidelines to a real context to assess their applicability. In addition, the discussions shed light on the practicality of those guidelines.

To ensure SecureDIS could be used by software engineers, formal methods of software verification were used. Chapter 6 addressed the experts' concerns in utilising the SecureDIS guidelines. The approach was to model the security policies of the SecureDIS. The process is a formalisation of the guidelines modelled with Event-B formal methods. Event-B was used for the verification along with the Rodin toolset. The results of the experiment show that the core of SecureDIS is implementable, by demonstrating the ability to transform the guidelines into mathematical invariants. In addition, the language used for the guidelines provides flexibility for software engineers to choose the techniques of their choice for implementation. These results answer the second research question.

To assess whether SecureDIS is practicable, through addressing the research question RQ3 in Section 3.1, a real data integration project was chosen. The case study employed

three data collection methods: interviews, questionnaires, and focus groups. The guidelines were addressed in light of the project selected, see Chapter 7. Despite the difficulty in finding an appropriate case to apply SecureDIS to, the study demonstrated that 78% of the framework is applicable. In addition, the project team members provided helpful feedback regarding the degree of usefulness of the guidelines for software engineers.

As a result, this research has shown the importance of building a secure DIS from the early stages and by considering the CPT properties to mitigate data leakage threats found in DIS. The use of proper techniques and the employment of security policies during the integration process has shown its links to the mitigation of the possible data leakage threats. The use of diverse research methods has shown that SecureDIS is applicable and implementable and accepted by software engineers.

8.2 Contributions

This research made the following contributions that can be beneficial to the research community:

1. A data leakage threat analysis of DIS with middle layers guided by a combination of the CPT properties. The guidance of the CPT properties is novel compared with that found in the existing literature. The analysis is based on two parts: the conceptualised components of DIS, and the anticipated leakage locations, both of which were confirmed by experts (Akeel et al., 2014). The analysis resulted in 25 different data leakage threats that can occur in a DIS. Since the analysis is property-driven, the information security academic domain can benefit from its steps.

2. A novel framework, SecureDIS, containing the architectural components of a DIS with middle layers addressing data leakage in a DIS environment. The framework components are based on the conceptualised DIS architecture that was confirmed by experts in the field. SecureDIS is guided by the concept of security by design through targeting software engineers to build secure software from its inception. (Akeel et al., 2015). Since SecureDIS is based on a real system abstraction, both of the information security and software engineering academic domains can use it for future research.

3. A set of 41 guidelines, within the SecureDIS framework, to mitigate the elicited threats of data leakage arranged by component. The guidelines are written in an informal language to assist software engineers to build secure DIS from the start (Akeel et al., 2013). The guidelines were validated by security experts and were assessed in terms of implementability, through formalisation, and applicability

through a case study. The guidelines can be used in software industry as they target system designers.

4. A novel security approach using Event-B formal method to assess the security policies within the DIS context. In a mathematical model, the security policies capture and formalise the SecureDIS guidelines. The model achieves information security assurance by checking the correctness and the consistency of the policies' design before implementation; therefore, it can be used in the software industry. It also ensures the coverage of the required security properties (Akeel et al., 2016). This can be useful to the information security academic domain as it provides a use case of building security policies.

5. The application of SecureDIS and its guidelines to a real data integration project through a case study, where security-related case studies are challenging to find and difficult to conduct. The ability to conduct security such case studies where information is confidential is a valuable aspect in the domain of information security. The study bridges the gap between theory and practice; hence, it is useful to both the software industry and the information security domains.

8.3 Future Work

The work described here can be used as a foundation for future research in the area of secure data integration, below are some of the proposed research directions.

Quantifying Data Leakage in DIS: One of the approaches to study data leakage is by modelling and investigating several real scenarios of data integration occurring in organisations, either in stand-alone systems, or in cloud integration services. A threat analysis can be conducted on the real scenarios to elicit data leakage threats relevant to the assets and the nature of the system. The results of the threat analysis can be combined with the historical attack data from the organisation's logs to provide more depth in the analysis. At this stage, the elicited data leakage threats can be used to build metrics quantifying the leakage impact on the assets before and after the application of a mitigation approach, such as the application of the SecureDIS guidelines. The real scenarios help to understand the risks of data exposure and its impact on the organisation.

Predicting Data Leakage Threats: By recording data consumers' profiles, the nature of queries to the DIS and other features, as well as the corresponding system responses to the queries, a data set can be created. This data set can be used as basis of analysing consumers' behaviour and predicting future actions. Using machine-learning algorithms from supervised learning can help in predicting possible data leakage threats to build a probabilistic model.

Standardising Data Integration Security: Many security standards are available, but are usually written at a high level and serve generic purposes. As far as is known, no data integration security standard is currently available. This work can be used to inform and shape new standards that focus on data integration, security, and system development.

A Tool for Checking the CPT Properties: The SecureDIS guidelines can be used in software tool to help software engineers in their task. The tool can include questions to assist software engineers determine which property was not covered by their DIS. The tool can provide a list of possible leakage threats that could arise from the failure to enforce a certain property.

Assessing Security Policies Implementation: Based on the discussions with experts, there seems to be a gap between the management who create the security policies and the developers who implement them. Future research can investigate bridging the gap between management and developers by analysing real studies and by proposing different approaches that can help upper management translate their requirements for the developers.

Event-B work can be extended in several directions: A first is to model an instance of a real DIS, along with its security policies, and check the correctness of those policies before deployment. Another direction is to use the correct model of the security policies to automatically generate the code for their enforcement, using the Rodin tool.

Appendix A

First Expert Reviews Material

The following is the material presented to the experts, in the first review to confirm the architectural components of the conceptualised DIS architecture.

Secure Data Integration Systems

1 Research Abstract

There is a need to guarantee accurate disclosure of private information when integrating data from disparate sources across different organisations varying in security and privacy requirements. This is achieved by maintaining those requirements and trusting other entities participating in the data integration process.

We aim to consider Security, Privacy, and Trust (SPT) as a whole to prevent information leakage from occurring by providing a novel framework that assists software engineers in designing such applications.

2 Objective

Utilising use cases taken from real life applications is needed to have valid contributions towards this research. Therefore, this document aims to validate a scenario adapted from (Bhowmick, Gruenwald, Iwaihara, *et al.*, 2006) to serve this purpose. Please read the scenario carefully and answer the questions on the last part of this document.

3 Scenario

To help understand the issues of having unsecure Data Integration Systems (DIS), we apply the concept to a UK healthcare domain.

Let's assume that a data integration application is required for a healthcare client, i.e. an application, to analyse an epidemiology problem. The purpose is to find the relationship between the types of death-causing diseases which decreased the population numbers in the UK in 2012. The results are then compared with public awareness of this epidemiology and the media's reaction.

At first, a user needs to have proper authorizations to complete any query and retrieve results. The process of the data integration includes: selecting data sources, integrating data, and finally presenting the results to an application.

Table 1 shows the details of each data source selected and the type of security level it has along with a label to be illustrated next in Figure 1.

Table 1: Data Sources Available for the Client, with Security Levels

Data Required	Purpose	Provided By	Security Level of Data Source	Possible Data Direction	Diagram Illustration
Diseases	To identify all spread diseases in the UK, choosing the location of interest, in the year 2012	NHS Information Centre	To authorised integrators only	NA	DS1
Census	To access the statistics to find the number of deaths in 2012 and cause of deaths	Office of National Statistics (ONS)	Open Government Data	From source to integrator	DS2
Public Awareness	To find out if the public are aware of the disease that causes early deaths	Media	Publically available	Possible both directions	DS3
Public Awareness	To know if the public are aware of the disease that causes early deaths	Social Networks [Twitter, Facebook]	Publically available, with limitations	Possible both directions	DS4

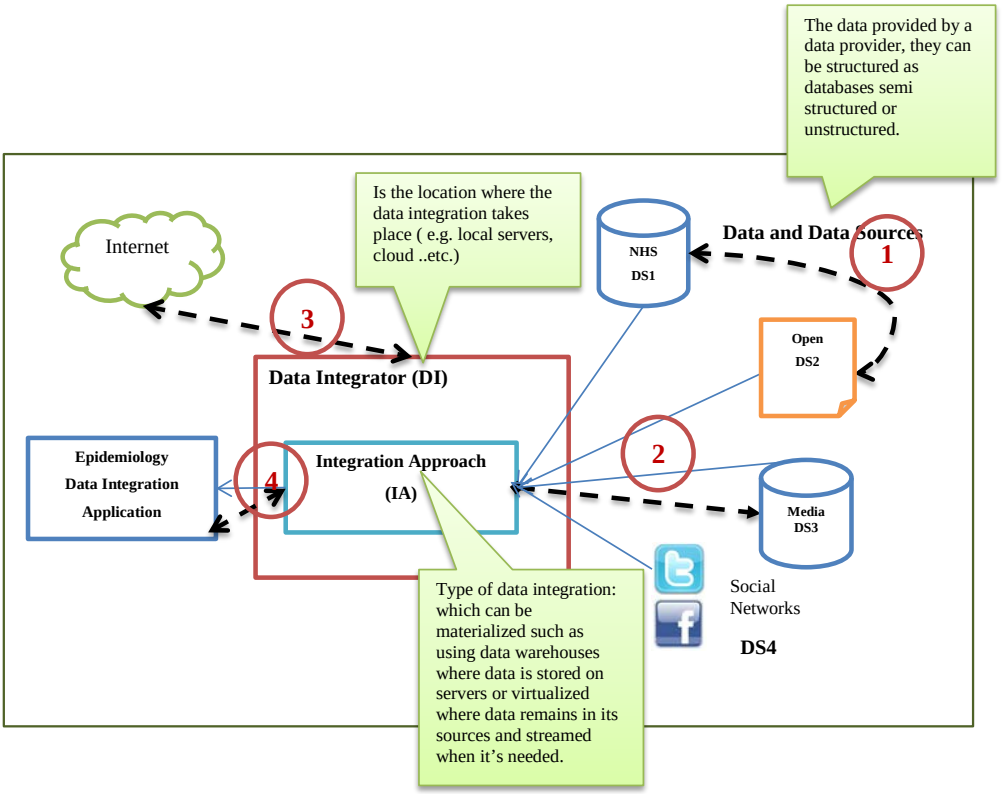

Figure 1: The Architecture of the Epidemiology Data Integration System

According to the previous scenario, there are many security, privacy, and trust issues that need to be considered. We summarise the questions arising from such a scenario in the following [Please note this is just to understand the problem, your response will be on section 5]:

In terms of **security**:

- How does the private data protect itself?
- How does the system continue to enforce security policies of the data sources during the integration?

In terms of **privacy**:

- Do data sources consider data anonymization?
- Is the data integration approach used considered as a privacy-preserving method?
- How does the trusted third party assisting in the integration comply with legislation and the organisation's policies?
- What is the privacy policy used by the applications accessing the integrated data?

In terms of **trust**:

- Are the data sources trustworthy?
- Is the server used for data integration secure and trusted?
- What trust mechanism is used to choose one data source over another?
- What trust mechanism is used to choose one data integrator over another?

4 The Scope of this Research

The previous security-related questions represent some of the issues that a DIS in this scenario may encounter.

This research focuses specifically on information leakage. The dashed arrows illustrated in Figure 1 are numbered and explained as follows:

1. A leakage between data sources as raw data.
2. A leakage that occurs between the results of the integration to some data source.
3. A leakage to the Internet, any other entity, either as a raw data or as a result of integration.
   Using third parties such as multi-tenancy public clouds to store and/or process data can be an important source of information leakage.
4. A leakage to users that may be caused by:
   - Using an access control model that allows leakage between overlapping roles and accidently granting permission to access private data.
   - Multiple consecutive queries to data sources (Clifton, Kantarcioğlu, Doan, *et al.*, 2004) or by inferring confidential information from un-confidential shared information or by inferring information from its statistical aggregates (Zhang, Zeng, Wang, *et al.*, 2011).

Therefore, there is a need for a list of security design requirements that considers these issues and assists software engineers in designing such a system. The proposed framework, namely *SecureDIS* aims to provide that considering security, privacy, and trust.

5 Expert Responses

We would like your feedback by validating the previous scenario according to the following:

- **From your experience, is this a reasonable scenario? ……**

- **Does it occur in real life applications? .....**

Type your comment here.

- **Do you have any recommendations of improvements? ……..**

Type your comment here.

- **Do you have any details that we should add to make a solid use case that helps in building a strong argument?**

Type your comment here.

- **If you have a clear application that does something similar, can you provide it with further details to cite it in this research? [It would be very beneficial and all intellectual property rights will be given]**

Type your comment here.

Assuming that:

- A framework provides a general/ broad customizable solution to this sort of problem.
- A model provides a solution to some aspect of the framework, showing how factors are related.
- An algorithm provides a detailed working solution to a specific part of the model.

What would do you recommend to be a proper practical and useful contribution to this problem (a framework, a model or an algorithm) and why?

Type your comment here

Many thanks for your time and cooperation

Appendix B

The Preliminary SecureDIS Guidelines

This appendix contains the preliminary SecureDIS guidelines before the confirmation by experts.

Table B.1: Guidelines for Mitigating Data Leakage in the Data and Data Sources Component

No.	**Guideline**	**DL Threat**	**C**	**P**	**T**
1	Check for security meta-data and privacy requirements	DL19	✓	✓	
2	Check for the purpose statement of the personal data used, as part of the privacy requirements	DL19		✓	
3	Based on the context of the DIS, remove identity attributes that can directly identify a person and add proper identity information that does not harm the privacy of an individual	DL25		✓	
4	Analyse data carefully to replace/remove attributes, such as QIDs, that have potential for inference by correlations or computations, using appropriate techniques	DL7, DL8, DL9		✓	
5	Create a security policy for each data source, based on security meta-data and privacy requirements, including the tolerated privacy loss	DL19	✓	✓	
6	Exclude small samples of sensitive data that can be recognised and inferred by deduction	DL21		✓	

Table B.2: Guidelines for Mitigating Data Leakage in the Security Policies Component

No.	Guideline	DL Threat	C	P	T
7	Ensure that the security policy considers the integrated security and privacy policies of the participating data sources and the DIS itself	DL15, DL16, DL17	✓	✓	
8	Define third parties' and public clouds' rights over data (including transitive trust)	DL11, DL12, DL13	✓	✓	✓
9	Define data providers' rights to access other data within the data integration system	DL20	✓	✓	✓
10	Define trust models used with system components and users	DL22, DL4			✓
11	Ensure data access and sharing is based on matching the purpose statement of both the data consumer and the data source privacy requirements	DL4		✓	
12	Ensure data access and sharing is based on the established trust	DL4	✓		✓

Table B.3: Guidelines for Mitigating Data Leakage in Data Consumers' Component

No.	Guideline	DL Threat	C	P	T
13	Resolve queries only when access to data is granted	DL1, DL2, DL3	✓		
14	Analyse the features of the requested queries, e.g. type of predicates, types of data returned, to identify possible security and privacy breaches	DL7, DL9	✓	✓	
15	Keep track of all queries and classify them according to 'type of threat' to prevent consecutive queries' inference attacks	DL6		✓	
16	Rewrite the queries after applying authorisation rules, privacy policies, and meta-data restrictions	DL18	✓	✓	
17	Protect query information, such as location and predicates	DL5		✓	

Table B.4: Guidelines for Mitigating Data Leakage in the Integration Approach Component

No.	Guideline	DL Threat	C	P	T
18	Before integration: Select suitable privacy-preserving data integration techniques based on query analysis	DL7,DL8, DL9		✓	
19	During integration: Encrypt data, in general, against the platform (i.e. integration location)	DL10	✓		
20	After integration: Analyse the query results to predict possible security and privacy policy violations and to determine further techniques to be applied before the result is returned to the consumer	DL7,DL8, DL9, DL25	✓	✓	
21	After integration: Compute the aggregated privacy loss of the integrated results using the privacy loss measure tolerated from each data source to satisfy privacy requirements	DL18		✓	
22	After integration: Annotate the results of the query with security and privacy meta-data and provide a clear description of their contents	DL19	✓	✓	

Table B.5: Guidelines for Mitigating Data Leakage in the Integration Location Component

No.	Guideline	DL Threat	C	P	T
23	Comply with security and privacy policy to prevent data leakage	DL18	✓	✓	✓
24	Obtain the data licence needed to access data sources	DL18, DL19		✓	
25	Establish trust with data providers to provide data sources	DL22			✓
26	In the case of outsourcing to an integration service, ensure the trustworthiness of the integration method-/service to integrate the data	DL11	✓	✓	✓
27	Protect the confidentiality of the data before and after the integration is achieved, (within the layers of the integration location)	DL10	✓		

Table B.6: Guidelines for Mitigating Data Leakage in the SSM Component

No.	**Guideline**	**DL Threat**	**C**	**P**	**T**
28	Conduct audits to ensure compliance with the DIS's security policies (including confidentiality, privacy, and trust)	DL15, DL16, DL17, DL24	✓	✓	✓
29	Configure the access control model used to include privacy needs	DL1, DL2, DL3	✓	✓	
30	Monitor and log successful and unsuccessful access to data (especially private data)	DL24	✓	✓	
31	In case where clouds are used to handle data (i.e. to access, process, store or manage data), establish the required trust to achieve those tasks	DL11, DL12, DL13	✓	✓	✓

Appendix C

Second Expert Reviews Material

This appendix contains the second expert reviews material including: contacting experts with an information sheet and a consent form, the reviews material, and a sample of a transcribed review.

C.1 Contacting Experts

Dear xxx.
My name is Fatmah; I am a PhD student in Computer Science at the University of Southampton, researching software engineering and security. I am writing to invite you to participate in an expert review to confirm and validate a set of guidelines that aim to prevent data leakage in data integration systems. The interview will include questions regarding your experience or knowledge of security, privacy, data management, and protection. The interview will be via Skype and it will be 45 minutes to one hour and it can be arranged at your convenient time. The results will help in the design of security guidelines and will identify areas of focus for further research in this area. I sincerely hope that you will consider participating in this review. I will be contacting you in the near future to confirm your interest in being interviewed. Please feel free to contact me with any questions. Please find the attached information sheet about this interview for your reference.
Sincerely,

The information sheet:

Participant Information Sheet- Version 2

Study Title: Security in Data Integration Systems

Researcher: Fatmah Akeel **Ethics number**: 8911

Please read this information carefully before deciding to take part in this research. If you are happy to participate you will be asked to sign a consent form.

What is the research about?
This research is required as part of the researcher's PhD degree in computer science. It focuses on security and privacy in data integration systems. The purpose of this research is to gain knowledge and understanding of how to secure data sharing and exchange in data integration systems using security guidelines. The questions asked are all directed towards security, privacy, and trust in data integration systems. This degree is fully funded by King Saud University, Riyadh Saudi Arabia.

Why have I been chosen?
You are invited to participate in this study because you are an expert in security and privacy, software engineering, or data management. Your opinion and expertise will help in creating effective guidelines that involve security, privacy and trust in data integration systems.

What will happen to me if I take part?
I will ask you to sign a consent form, and then the study will begin. I will conduct an interview with you, with open-ended questions, and I will record your voice during the interview.

Are there any benefits in my taking part?
This research is not designed to help you personally, but your feedback will help me gather expert opinions in the area of research.

Are there any risks involved?
No

Will my participation be confidential?
Yes. Any data will be stored and will not be linked to your name or to your organisation's name. Your data and that of other participants will be stored and used on secure systems. Any information related to your organisation will not be disclosed. Your organisation name will be mentioned as a government agency, a private company or a research institute only.

What happens if I change my mind?
You have the right to terminate your participation in the research, at any stage, you do not need to give any reasons, and without your legal rights being affected. Any data collected from you will be immediately destroyed.

What happens if something goes wrong?
In the unlikely case of concern or complaint, please contact Dr Martina Prude, Head of Research Governance (02380 595058, mad4@soton.ac.uk).

Where can I get more information?
For further details, please contact either myself or my study supervisors.
Fatmah Akeel: fya1g12@ecs.soton.ac.uk
Dr. Gary B. Wills: gbw@ecs.soton.ac.uk
Dr. Andrew M. Gravell: amg@ecs.soton.ac.uk

The consent form:

CONSENT FORM (*Version 2*)

Study title: Secure Data Integration Systems

Researcher name: Fatmah Akeel
Study reference: ERGO/FoPSE/8911
Ethics reference: 8911

Please initial the box (es) if you agree with the statement(s):

I have read and understood the information sheet (version 2, 10/7/2014) and have had the opportunity to ask questions about the study.	☐
I agree to take part in this research project and agree for my data to be used for the purpose of this study	☐
I understand my participation is voluntary and I may withdraw at any time without my legal rights being affected	☐
I am happy to be contacted regarding other unspecified research projects. I therefore consent to the University retaining my personal details on a database, kept separately from the research data detailed above. The 'validity' of my consent is conditional upon the University complying with the Data Protection Act and I understand that I can request my details be removed from this database at any time.	☐

Data Protection
I understand that information collected about me during my participation in this study will be stored on a password protected computer and that this information will only be used for the purpose of this study. All files containing any personal data will be made anonymous.

Name of participant (print name)……

Signature of participant……..

Date……

C.2 Reveiws Material

The following are the material used in the **second expert reviews** to confirm SecureDIS components and guidelines.

Expert Reviews to Confirm SecureDIS

- Please **specify the number of years** of experience, in the fields that best describe your area of expertise. You can fill more than one area of expertise.

Expert's area of Expertise

	Company (Business)	Government Organisation	Researcher
Security and Privacy	[ ]	[ ]	[ ]
Software Engineering (System analysis)	[ ]	[ ]	[ ]
Databases/ data related speciality	[ ]	[ ]	[ ]

If you cannot find you area of expertise please indicate it here ……..

- What type of projects do you currently work on, you can select more than one:
  - Data management / administration
  - Data integration
  - System development
  - Security management
  - Research
  - Other …….

SecureDIS Framework

Figure 1 illustrates the components of SecureDIS, a framework that aims to secure Data Integration Systems (DIS) considering Security, Privacy, and Trust to prevent data leakage.
What do we mean by security? It is the combination of confidentiality, integrity, and availability. The attribute of concern is confidentiality other attributes are assumed to be implemented. Confidentiality is achieved by limiting access to data to authorised individuals, entities and process.
Privacy: is concerned with protecting personal information.
Trust: is the belief that an entity will behave in a predictable manner by following a security policy.
Data leakage: is disclosing private information intentionally or unintentionally to unauthorised parties.

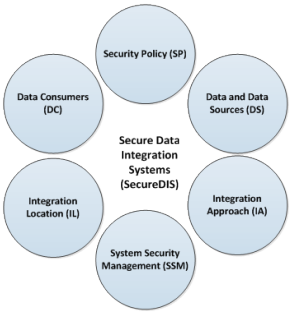

Figure 1: SecureDIS Framework

SecureDIS is an architectural framework to design secure DIS.

***Target Audience*:** security system analysts

Purposes and Uses: can be used as:

- Design guidelines
- A qualitative evaluation checklist

Data Integration System Architecture

Figure 2 illustrates DIS architecture for generic applications, based on SecureDIS framework, that aim to integrate heterogeneous data coming from different sources to resolve data consumer queries.

Assumptions:
It is assumed that the integration process needs a middle layer to achieve the integration. In addition, it is assumed that data consumers', i.e. users, access rules are already defined based on their roles.

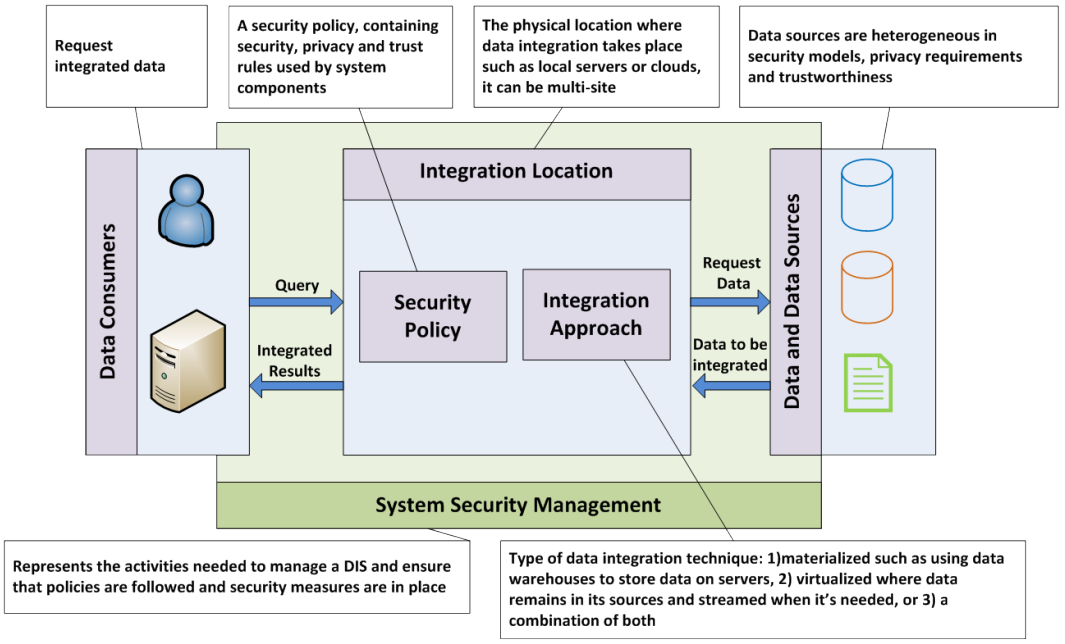

Figure 2: DIS architecture based on SecureDIS

DIS Architecture and Data Leakage

Figure 3 illustrates the DIS architecture with data leakage locations, each location may have one or more threats, explained in Table 1.

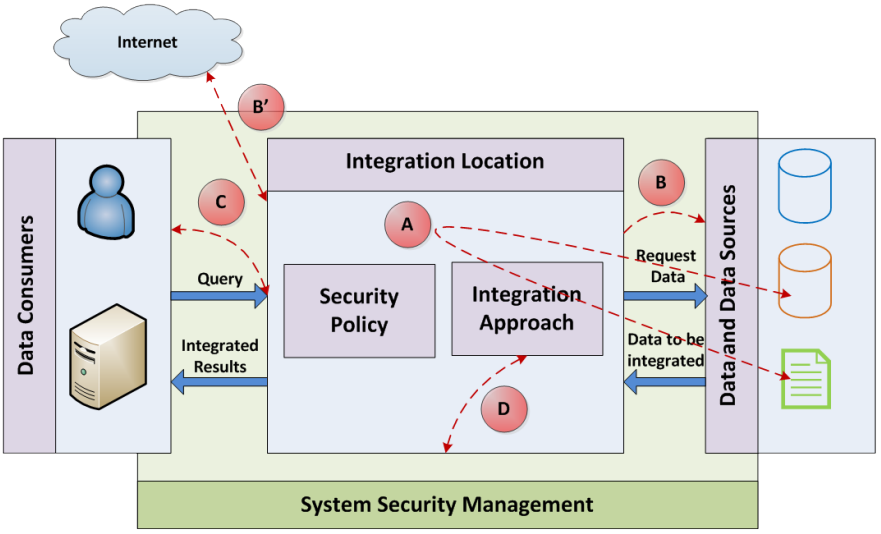

Figure 3: DIS Architecture with Data Leakage Locations

Table 1: Data Leakage Threats Explanation, based on Figure 3

Threat Location	Threat No.	Type[1]	Threat Type
A	**A.1**	**P**	Leakage between data providers by inference attacks based on accessing pieces of data (Goryczka, Xiong & Fung, 2013).
B,B'	**Data leakage between the Integration Location and Data Sources or the Internet**		
	B.1	**CPT**	Using public clouds to process, store, integrate private data (Carey, Onose & Petropoulos, 2012) (Ristenpart, Tromer, Shacham, *et al.*, 2009)
	B.2	**CPT**	Lack of security policies for transitive trust from one party to other parties (Fung, Trojer, Hung, *et al.*, 2012).
	B.3	**CP**	Unclarified third parties rights on data (Meingast, Roosta & Sastry, 2006).
	B.4	**C**	Violation of security policies due to inapplicable confidentiality on merged data (Batty, Crooks, Hudson-Smith, *et al.*, 2010).
	B.5	**C**	Violation of security policies due to ignorance of legal issues on data management (Batty, Crooks, Hudson-Smith, *et al.*, 2010).
	B.6	**C**	Violation of confidentiality due to inconsistent regulatory laws (Meingast, Roosta & Sastry, 2006).
C	**Data leakage caused from Data Consumers side**		
	C.1	**C**	**Unauthorised access:** caused by inappropriate choice of access control model(Braghin, Cortesi & Focardi, 2003), access control weakness (Tipton, 1998)or misconfigured access control (Pistoia, Fink, Flynn, *et al.*, 2007).
	C.2	**P**	**Inference Attack (consecutive queries):** use of multiple queries to infer information (Bhowmick, Gruenwald, Iwaihara, *et al.*, 2006).
	C.3	**P**	**Inference Attack 1) Record Linkage:** use of un-confidential information and statistical aggregates to infer private data (Clifton, Kantarcioğlu, Doan, *et al.*, 2004) (Zhang, Zeng, Wang, *et al.*, 2011). **2) Attribute-linkage:** Linking QID attributes together (Fung, Trojer, Hung, *et al.*, 2012) or with external data (Whang & Garcia-Molina, 2012).
	C.4	**P**	**Inference Attack (Query Attribute-correlation):** gathering information about queries predicates and correlating them to infer sensitive information about the owner (Li, Luo, Liu, *et al.*, 2013).
	C.5	**P**	**Inference Attack (interval disclosure):** computing the missing values using already published data (Boyens, Krishnan & Padman, 2004).
D	**D.1**	**C**	Leaking the integrated results to the Integration Location by lack of data encryption against the platform (Herbert & Thieme, 2012).

Other generic threats to:

P. Privacy Threats	
P.1	Revealing Personally Identifiable Information (PII).
P.2	Using personal and sensitive data for unintended purposes.
P.3	Generic inference attacks that violate privacy of individuals, includes P.1, P.2, C.2, C.3, C.4, C.5.
T. Trust Threats	
T.1	A threat caused by behaving against the agreed and stated security policy.

[1] C = Confidentiality, P=Privacy, T=Trust

Theme 1: About your Current Practices

Q1: What sort of security guidelines, standards and best practices related to data security are used currently in your organisation? …… …… **Q2: what would you think the reason for this choice of guidelines that you use or that others use in the domain of data security?** …… ……

Q3: Looking at the previously discussed data leakage threats, do you have any more threats to include?	Yes	No
If yes, please explain the threats:		
Q4: Are you aware of any data leakage threats in your organisation? [please note it's a yes or no question]	Yes	No
Q5: If yes, have you implemented any countermeasures against those threats?	Yes	No
If not, please explain the reasons:		

Theme 2: Validating SPT Activities to Overcome Data Leakage Threats

The following are SecureDIS guidelines, categorised according to the previous architecture components. These guidelines aim to overcome data leakage threats in DIS.
Please validate the following guidelines assuming that we are going to design the previous DIS in-house (i.e. we are not comparing current DIS guidelines with existing technologies). In addition, feel free to include any other activities that are most likely relevant to overcome data leakage and that are useful in real-life applications (or in your organisation).

1- Data and Data Sources Component

No	Guidelines	To Mitigate	Strongly Disagree	Disagree	Neither agree nor disagree	Agree	Strongly Agree
Before accepting data and data sources:							
1	**Check** for *security meta-data* and privacy requirements	C.1 P.3					
2	**Check** for the *purpose statement* of the used personal data as part of the privacy requirements	P.2					
3	Based on the context of DIS, **Remove** *identity attributes* that can directly identify a person and add proper identity information that does not harm the privacy of an individual	P.1					
4	**Analyse** data carefully to replace/remove attributes, such as QIDs, that has a potential of inference by correlations or computations, using techniques such as: generalization, suppression and perturbation	C.3 C.5					
5	**Create** a *security policy* for each data source based on security meta-data and privacy requirements including the tolerated *privacy loss*	C.1 P.3					
6	**Do not include** small samples of sensitive data that can be recognized and inferred by deduction	C.3					

If you have disagreed on any of the previous guidelines, can you elaborate why?
……
……

Q6: From your experience, which of the following can be used as criteria of trustworthiness for both data and data sources? [You can select more than one]

- Update-frequency of a data source (data freshness)
- The frequency of data source use by other entities
- Certification by some trusted entity
- Other , specify

……
……

2- Security Policy Component

No	Guidelines	To Mitigate	Strongly Disagree	Disagree	Neither Agree Nor disagree	Agree	Strongly Agree
7	**Ensure** that the security policy considers the integrated security and privacy policies of the participating data sources and the DIS itself	B.4					
8	**Define** third parties and public clouds rights on data (including transitive trust)	B.2 B.3					
9	**Define** data providers rights to access other data within the data integration system	A.1					
10	**Define** trust models used with system components and users	T.1					
11	**Ensure** data access and sharing is based on matching the *purpose statement* of both the data consumer and the data source privacy requirements	P.2					
12	**Ensure** data access and sharing is based on the established trust	C.1 T.1					

If you have disagreed on any of the previous guidelines, can you elaborate why?

……
……

3- Data Consumers Component (Queries)

Data consumers request data by queries, the following are specific to handling the query that will be processed in the Integration Location.

No	Guidelines	To Mitigate	Strongly Disagree	Disagree	Neither agree nor disagree	Agree	Strongly Agree
13	**Resolve** queries only when *access to data* is granted	C.1					
14	**Analyse** the features of the requested queries e.g. type of predicates, types of data returned to know possible security and privacy breaches	C.4					
15	**Keep track** of all queries and classify them according to 'type of threat' to prevent consecutive queries inference attacks	C.2					
16	**Rewrite** the queries after applying authorization rules, privacy policies, and meta-data restrictions	C.1 P.3					
17	**Protect** query information such as location and predicates	C.4					

If you have disagreed on any of the previous guidelines, can you elaborate why?

……
……

4- Integration Approach Component

No	Guidelines	To Mitigate	Strongly Disagree	Disagree	Neither agree nor disagree	Agree	Strongly Agree
18	**Before the integration:** Select suitable privacy preserving data integration techniques based on query analysis	C.3 C.5					
19	**During the integration:** Encrypt data, in general, against the platform (i.e. integration location)	D.1					
20	**After the integration:** Analyse the query results to predict possible security and privacy policy violations and to determine further techniques to be applied before the result is returned to the consumer	C.3 C.5					
21	**After the integration:** Compute the aggregated privacy loss of the integrated results using the tolerated privacy loss measure from each data source to satisfy privacy requirements	P.3					
22	**After the integration:** Annotate the results of the query with security and privacy meta-data and provide a clear description of their contents	C.1 P.3					

If you have disagreed on any of the previous guidelines, can you elaborate why?

……

……

5- Integration Location Component

No	Guidelines	To Mitigate	Strongly Disagree	Disagree	Neither agree nor disagree	Agree	Strongly Agree
23	**Comply** with security and privacy policy to prevent data leakage	B.5 B.6					
24	**Obtain** the needed data licence to access data sources	P.2 C.1					
25	**Establish** trust with data providers to provide data sources	T.1					
26	In case of outsourcing to an integration service, **ensure** the trustworthiness of the integration method/service to integrate the data	B.1 T.1					
27	**Protect** the confidentiality of the data before and after the integration is achieved, (within the layers of the integration location)	C.1					

If you have disagreed on any of the previous guidelines, can you elaborate why?

……

……

6- System Security Management Component

No	Guidelines	To Mitigate	Strongly Disagree	Disagree	Neither agree nor disagree	Agree	Strongly Agree
28	**Conduct** audits to ensure compliance with the DIS's security policies (including confidentiality, privacy and trust)	B.4 B.5 B.6					
29	**Configure** the used access control model to include privacy needs	C.1 P.2					
30	**Monitor and log** successful and unsuccessful access to data (especially private data)	C.1					
31	In case of using clouds to handle data (i.e. to access, process, store or manage data), **establish** the required trust to achieve those tasks	B.1					

If you have disagreed on any of the previous guidelines, can you elaborate why?

……

……

Q7: Reading the guidelines, do you find them similar to other standards, guidelines, and best practices that you know?	Yes	No
If yes please elaborate:		
Q8: Are there anymore aspects of DIS that were not considered by SecureDIS?	Yes	No
If yes, please elaborate:		

Q9: Do you think that:					
a) SecureDIS covers data **confidentiality**	Strongly Disagree	Disagree	Neither agree nor disagree	Agree	Strongly Agree
b) SecureDIS covers **data privacy**	Strongly Disagree	Disagree	Neither agree nor disagree	Agree	Strongly Agree
c) SecureDIS covers **trust** within DIS components	Strongly Disagree	Disagree	Neither agree nor disagree	Agree	Strongly Agree
d) SecureDIS **components** are suitable	Strongly Disagree	Disagree	Neither agree nor disagree	Agree	Strongly Agree

If you have disagreed on any of the previous statements, can you elaborate why?

……

……

C.3 A Sample of a Review

The following is a sample of a transcribed expert's review, from the second expert reviews.

Expert's area of Expertise

	Company (Business)	Government Organisation	Researcher
Security and Privacy	[16]	[ ]	[7]
Software Engineering (System analysis)	[28]	[ ]	[13]
Databases/ data related speciality	[28]	[ ]	[7]

Theme 1: About your Current Practices
Q1: What sort of security guidelines, standards and best practices related to data security are used currently in your organisation?
We are using COBIT 5 Framework as our security guidelines; we are planning to be certified for ISO 27000 next year
Q2: Is there a particular reason for this choice of guidelines?
We are using COBIT for the past 8 years.

Q3: Looking at the previously discussed data leakage threats, do you have any more threats to include?	**Yes**	
If yes, please explain the threats: There are major DL threats from our IT technology providers (hardware, and software), like backdoors, batches, support. This is a major risk and it need to be addressed as part of an overall Risk Management Program, under the section Vendor Management. Remember, IT suppliers have their connections to their government security agencies. Case in point NSA spying on everyone.		
Q4: Are you aware of any data leakage threats in your organisation?	**Yes**	
Q5: If yes, have you implemented any countermeasures against those threats?	**Yes**	
We are planning to establish an Enterprise Risk Management (ERM) system for the whole company, which will include a vendor management, that will ensure that we do not have an invisible Elephant within our IT operations or infrastructures components		

Q6: From your experience, which of the following can be used as criteria of trustworthiness for both data and data sources? [You can select more than one]

- Update-frequency of a data source (data freshness)
- The frequency of data source use by other entities
- Certification by some trusted entity
- Other , specify

I will select Certification by some trusted entity, because that will ensure that proper due diligence has been made prior to the issuance of this certification. Also, it will be protecting the firm from any liability is the future, if any legal case arise. No Data should be accepted without certification or a waiver for not been certified.

Q7: Reading the guidelines, do you find them similar to other standards, guidelines, and best practices that you know?	**Yes**	
If yes please elaborate: These DIS guidelines represent a holistic approach for the first time, in order to minimize Data Leakage Threats. There are single points of fragmented guidelines available, but it's cumbersome and need huge efforts for their users to aggregate them and make send of all of them.		

Appendix D

Case Study Material

This appendix contains the case study material including: the organisation's approval to conduct the study, the emails sent to participants, the results coming from the degree of compliance questionnaire (stage 2), the coding details used in stage 3, the applicable vs. applied analysis conducted as part of stage 3, and the usefulness questionnaire used in stage 4.

D.1 The Organisation's Approval

The email shown below was sent to the organisation to seek their approval to conduct the case study, it is anonymised for data protection.

> Dear Mr. xxx,
> My name is Fatmah Akeel, a Computer Science PhD student at the University of Southampton, UK. My research topic is between software engineering and information security. I have developed an architectural framework that contains a set of guidelines that can be used by organisations employing data integration to reduce possible threats of data leakage during the design of systems. (More information about my work can be found here `http://users.ecs.soton.ac.uk/fya1g12/`) I have discussed my research with XX, CC'ed, and he explained to me that you may have a project in your department with a similar nature. Therefore, I am contacting you today to discuss the possibility of conducting a case study research on your project which will be very suitable for my PhD. As a result, I think this is a beneficial opportunity to your project as the case study will provide a useful assessment focusing on confidentiality, privacy, and trust aspects of the project.

If you agree on taking part in this study, the results of the analysis of this assessment will be provided as a report to you with suggested improvements. In addition, the experience of conducting a case study will create an opportunity for the employees to get exposed to other dimensions of research and how it can be useful to improve an organisation. Your help in this study will be a contribution to the body of knowledge, and it is a good opportunity to be part of a scientific research. Finally, this case study will assist me personally in completing my PhD by evaluating the framework on a real life application.
In order to conduct this study successfully, I need your initial acceptance before I start designing the case study. It is planned to include several questionnaires and interviews with employees involved in the project in addition to observations. And the proposed time to conduct the study is approximately 4 to 6 weeks, using 2 to 4 hours per week of working time. I would like you to know that I am willing to deal with data in a confidential manner and sign any required data secrecy and privacy agreements to conduct this study. Also, the reports resulting from the study will be forwarded to you for approval before using them in my thesis. I sincerely hope that you consider this opportunity which will be useful for both your department and myself. Please do not hesitate to contact me any time to discuss this further.
Regards,
Fatmah

D.2 Emails to Participants

The following are the emails sent to participants to take part in the questionnaires (Stage 2).

14/5/2015
Dear Participant,
Thank you for your participation in my case study. As we discussed, the following link contains a questionnaire regarding the compliance of your team with the proposed framework.
Here is the link: `https://www.isurvey.soton.ac.uk/14565`
Many thanks for your cooperation.
Fatmah

30/5/2015
Dear xx,
I hope this message finds you well. As you know, with your help and support, we have completed the first stage of the case study successfully. And we need to complete the second stage for the case study to be valid. Therefore, this is a gentle reminder for you to kindly complete the online survey that you have started. Here is the link: `https://www.isurvey.soton.ac.uk/14565`. Please note that your participation is very valuable for this study to be successful, and I am looking forward for your response. You can always contact me with any issues with the survey.
Many thanks for your time.
Regards, Fatmah

D.3 Initial Applicability Responses

Table D.1 shows the data collected from the participants on the initial applicability of the SecureDIS guidelines to the data integration project. The column labelled Response is collected from the questionnaires (Stage 2). The results were revisited in the focus group meeting (Stage 3) and the changes are shown in the column labelled Final Response. The values are as follows:

- SD = Strongly Disagree
- D= Disagree
- N= Neutral
- A = Agree
- SA= Strongly Agree
- N/A = Not applicable

Table D.1: Participants Responses to the Questionnaires (Stage 2)

Guideline	Response	Final Response	Guideline	Response	Final Response
1	A, D	A	**22**	SA, N	
2	A, D	A	**23**	SA	
3	N, D		**24**	SA, A	A
4	N, N	N/A	**25**	A, N	
5	N, N	A	**26**	SA, D	D
6	N, D		**27**	N, D	
7	N, SD		**28**	A, N	
8	N, D	A	**29**	SA, A	
9	A, A		**30**	SA, A	
10	SA, SA	A	**31**	SA, A	
11	SA ,SA	A	**32**	SA, A	D
12	SA, A		**33**	SA, SA	N/A
13	SA, A		**34**	SA, A	
14	SA, A		**35**	SA, N	
15	A, A		**36**	A, N	
16	A, N		**37**	A, N	
17	SA, N		**38**	SA, N	
18	A, A		**39**	SA, SA	
19	SA, A		**40**	SA, N	
20	SA, A		**41**	SA, A	
21	A, N				

D.4 Coding Details

The codes of SecureDIS guidelines and the participants feedback on the first stage are shown in Tables D.2 to D.5.

Table D.2: Linking Codes (Nodes) with Stage 1 and SecureDIS Guidelines

Category	**Main Node**	**Sub Node**	**Analysis Themes (Stage 1)**	**No of Guidelines**	**SecureDIS Guidelines**
I.DIS	Project Nature	General	Project Nature, Data Sources and Integration Process, Security and SDLC	0	None
	Data Sources	General	Data Sources and Integration Process	8	1, 2, 3, 4, 5, 6, 7, 8
	Data Sources	Data Providers	Project Nature, Data Sources and Integration Process	6	7, 8, 12, 14, 31, 32
	Security Policy	General	Security Models, Security Policy, Analysis Phase, Design Phase	15	5, 10, 11, 12, 13, 14, 15, 16, 17, 18, 20, 26, 30, 36, 37
	Security Policy	Several Security Policies	Security Models	1	9

Table D.3: Linking Codes (Nodes) with Stage 1 and SecureDIS Guidelines - continued

Category	**Main Node**	**Sub Node**	**Analysis Themes (Stage 1)**	**No of Guidelines**	**SecureDIS Guidelines**
I.DIS	External Entities	General	Project Nature	2	10,11
	Integration Approach	General	Data Sources and Integration Process	6	24, 25, 26, 27, 28, 29
	Integration Location	General	Data Sources and Integration Process	6	30, 31, 32, 33, 34, 35
	Data Consumer	General	Data Sources and Integration Process, Security Model, Security Policy, Implementation Phase, Testing Phase	9	14, 15, 19, 20, 21, 22, 23, 26, 29
	System Security Management	General	None	12	16, 21, 29, 30, 31, 35, 36, 37, 38, 39, 40, 41

Table D.4: Linking Codes (Nodes) with Stage 1 and SecureDIS Guidelines - continued

Category	**Main Node**	**Sub Node**	**Analysis Themes (Stage 1)**	**No of Guidelines**	**SecureDIS Guidelines**
II. Properties	Confidentiality	General	Security Models	14	1, 5, 9, 20, 21, 23, 26, 28, 29, 30, 31, 34, 36, 37
	Confidentiality	Encryption	Trust Models, Implementation Phase	2	22, 25
	Confidentiality	Access Control	Implementation Phase	9	10, 12, 14, 15, 19, 38, 39, 40, 41
	Privacy	General	Security Models	12	5, 8, 20, 21, 24, 26, 27, 28, 30, 36, 37, 38
	Privacy	Anonymity	None	3	3, 4, 6
	Privacy	Data Use	Data Sources and Integration Process, Security Policy	3	2, 14, 31
	Trust	General	Trust Models, Implementation Phase	10	7, 10, 11, 13, 15, 18, 32, 33, 36, 37
	Authentication	General	Data Sources and Integration Process, Implementation Phase	0	None

Table D.5: Linking Codes (Nodes) with Stage 1 and SecureDIS Guidelines - continued

Category	**Main Node**	**Sub Node**	**Analysis Themes (Stage 1)**	**No of Guidelines**	**SecureDIS Guidelines**
III. Security Guidance	General	-	Security Policy, Security Guidance, Implementation Phase, Testing Phase	0	None
IV. Security and SDLC	General	-	Security and SDLC	0	None
	Analysis Phase	Analyse Attacks General	Analysis Phase	16	1, 2, 4, 6, 7, 10, 11, 12, 13, 15, 16, 18, 20, 30, 32, 33
	Design Phase	Apply Techniques General	Design Phase	25	1, 2, 3, 4, 5, 8, 9, 11, 14, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 32, 33, 34, 37, 38, 39
	Implementation Phase	General	Implementation Phase	5	24, 25, 26, 27, 28
	Testing Phase	General	Testing Phase	5	9, 14, 17, 23, 40

D.5 Usefulness Questionnaire

The usefulness questionnaire assess SecureDIS in three parts. The first part focused on the evaluation of the overall comprehensiveness and benefits of the framework. The second part was to assess the usefulness in particular, the elements used in this part were adapted from the work of Davis (1989). Each statement of the usefulness survey was presented to the participants with a 5-point Likert scale,to give the team members more choice to indicate the agreement to the statements describing SecureDIS qualities (Likert, 1932). The scale includes 'Strongly Disagree','Disagree','Not Sure', 'Agree','Strongly Agree'.

D.5.1 SecureDIS Qualities Questionnaire

The following questionnaire was given to the case study's participants during the focus group. Participants were asked to respond to each statement.

Table D.6: SecureDIS Qualities Questionnaire

Statement	**Strongly Disagree**	**Disagree**	**Not Sure**	**Agree**	**Strongly Agree**
About SecureDIS Framework:					
SecureDIS is comprehensive in terms of covering DIS components with a middle layer.					
SecureDIS is comprehensive in terms of covering confidentiality.					
SecureDIS is comprehensive in terms of covering privacy.					
SecureDIS is comprehensive in terms of covering trust.					
SecureDIS guidelines can be adapted to different contexts.					
The level of detail provided by SecureDIS guidelines is suitable for system analysts.					
Using SecureDIS helps reducing data leakage threats.					
SecureDIS guidelines are implementable.					
SecureDIS guidelines are practical.					

Table D.7: SecureDIS Qualities Questionnaire- continued

Statement	**Strongly Disagree**	**Disagree**	**Not Sure**	**Agree**	**Strongly Agree**
Usefulness Assessment:					
Using SecureDIS framework in my job would enable me to accomplish tasks more quickly.					
Using SecureDIS framework would improve my job performance.					
Using SecureDIS framework in my job would increase my productivity.					
Using SecureDIS framework would enhance my effectiveness on the job.					
Using SecureDIS framework would make my job easier.					
Overall, I would find SecureDIS framework useful in my job.					

D.5.2 SecureDIS Qualities Questionnaire Results

The following charts show the participants responses to each part of the questionnaire. Figure D.1 shows the comprehensiveness of SecureDIS in terms of the CPT properties. Figure D.2 shows other SecureDIS qualities. Figure D.3 shows the assessed elements of the SecureDIS usefulness where:

- A: Accomplishes Tasks
- B: Improves Performance
- C: Increases Productivity
- D: Enhances Effectiveness
- E: Makes Job Easier
- F: Overall Useful

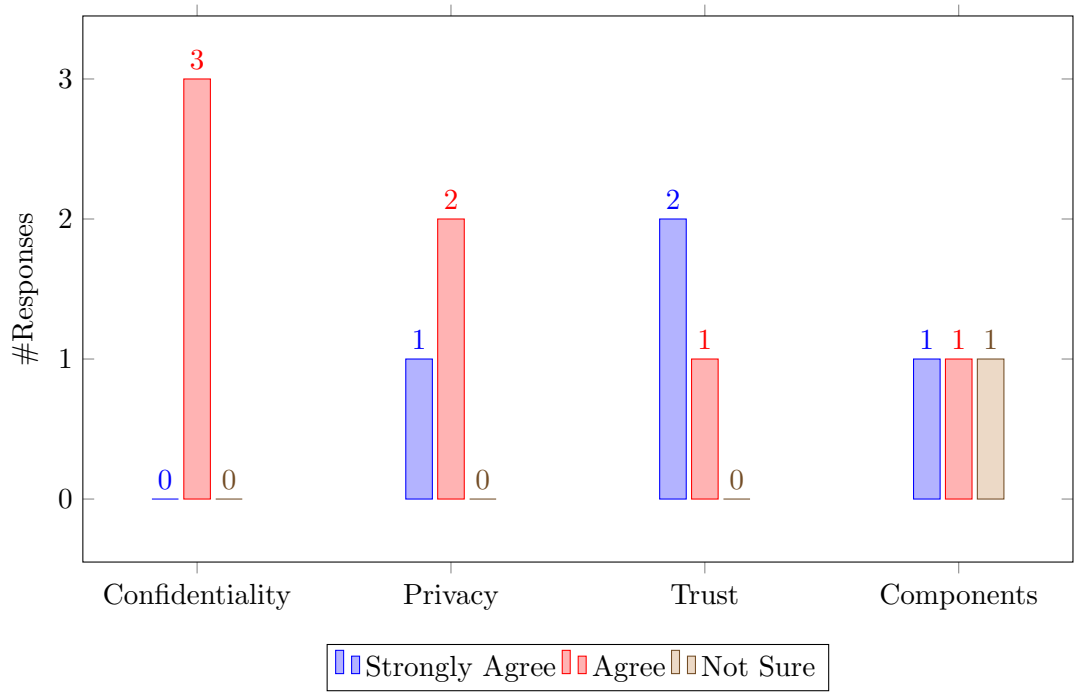

Figure D.1: The Comprehensiveness of SecureDIS

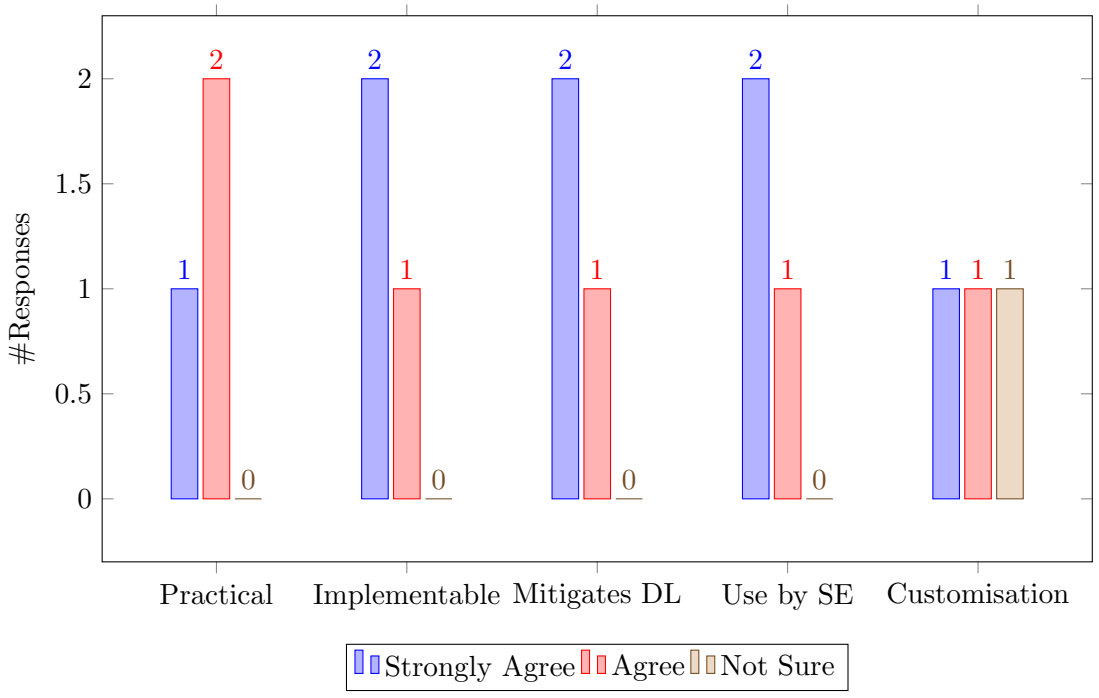

Figure D.2: Other SecureDIS Qualities

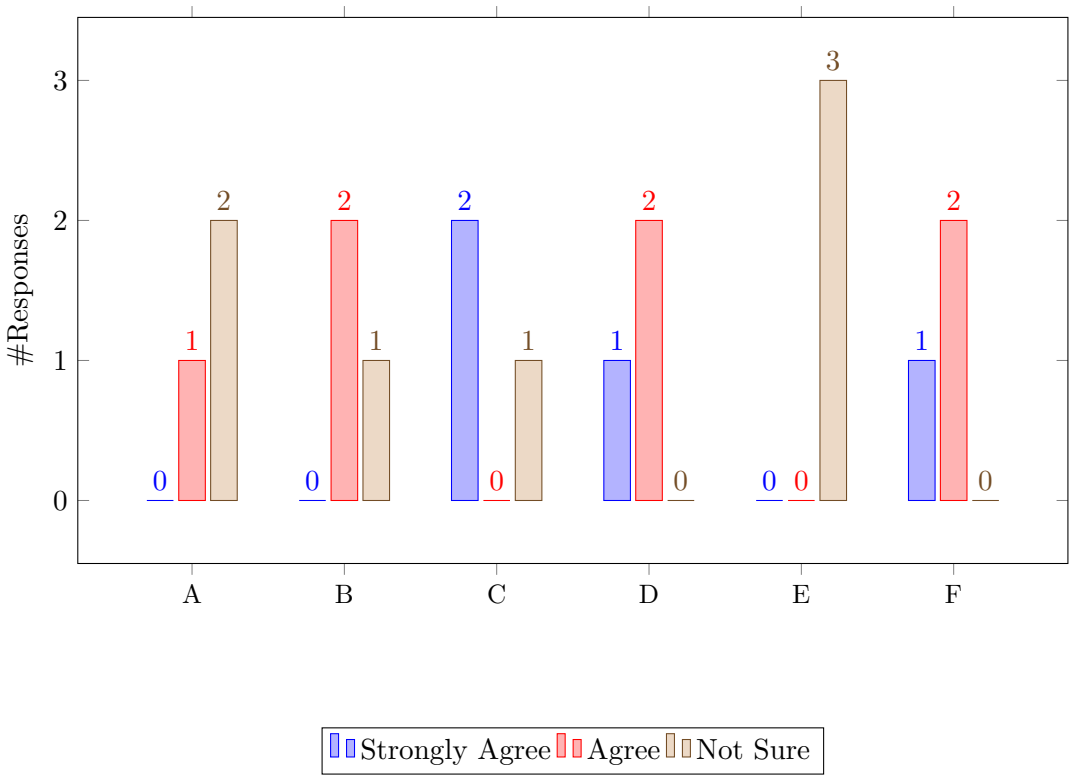

Figure D.3: SecureDIS Usefulness to the Project

References

Aagedal, J. Ø., Braber, F. D., Dimitrakos, T., Gran, B. A., Raptis, D., and Stølen, K. (2002). Model-based Risk Assessment to Improve Enterprise Security. In *Proceedings of the Sixth International Enterprise Distributed Object Computing Conference (EDOC'02)*, Oslo, Norway. IEEE.

Abrial, J. R. (2010). *Modeling in Event-B: System and Software Engineering*. Cambridge University Press, New York, NY, USA, 1st edition.

Abrial, J. R., Butler, M., Hallerstede, S., and Voisin, L. (2006). *An Open Extensible Tool Environment for Event-B*, pages 588–605. Springer Berlin Heidelberg, Berlin, Heidelberg.

Agudo, I., Fernandez-Gago, C., and Lopez, J. (2010). A scale based trust model for multi-context environments. *Computers and Mathematics with Applications*, 60(2):209–216.

Ahmed, R., De Smedt, P., Du, W., Kent, W., Litwin, M. A. K. W. A., Rafii, A., and Shan, M. C. (1991). The Pegasus heterogeneous multidatabase system. *Computer*, pages 19–27.

Akeel, F., Salehi Fathabadi, A., Paci, F., Gravell, A., and Wills, G. (2016). Formal Modelling of Data Integration Systems Security Policies. *Data Science and Engineering*, 1(3):139–148.

Akeel, F., Wills, G., and Gravell, A. (2013). SecureDIS: a Framework for Secure Data Integration Systems. *The 8th International Conference for Internet Technology and Secured Transactions*, pages 588–593.

Akeel, F. Y., Wills, G. B., and Gravell, A. M. (2014). Exposing Data Leakage in Data Integration Systems. *The 9th International Conference for Internet Technology and Secured Transactions (ICITST-2014)*, pages 420–425.

Akeel, F. Y., Wills, G. B., and Gravell, A. M. (2015). Secure Data Integration Systems. In *5th International Conference on Cloud Computing and Services Science - Doctoral Consortium*, pages 26–37, Lisbon, Portugal.

Anderson, R. (1996). A security policy model for clinical information systems. In *IEEE symposium on Security and Privacy*, pages 30–43, Oakland, CA, USA.

Apvrille, A. and Pourzandi, M. (2005). Secure Software Development by Example. *IEEE Security and Privacy Magazine*, 3(4):10–17.

Arenas, A. E., Aziz, B., Bicarregui, J., and Wilson, M. D. (2010). An Event-B approach to data sharing agreements. In *Integrated Formal Methods-LCNCS*, volume 6396, pages 28–42.

Artz, D. and Gil, Y. (2007). A survey of trust in computer science and the Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):58–71.

Bahga, A. and Madisetti, V. K. (2015). Healthcare data integration and informatics in the cloud. *Computer*, 48(2):50–57.

Balci, O. (1994). Validation, verification, and testing techniques throughout the life cycle of a simulation study. In *Proceedings of the 26th conference on Winter simulation WSC '94*, number 1987, pages 215–220, San Diego, CA, USA.

Balouziyeh, J. M. B. and Husein, A. T. (2012). The Legal Framework for Privacy and Data Protection in Saudi Arabia. Available at `http://www.americanbar.org/publications/international_law_news/2012/fall/legal_framework_privacy_data_protection_saudi_arabia.html`.

Barhamgi, M., Benslimane, D., Ghedira, C., and Gancarski, A. L. (2011). Privacy-Preserving Data Mashup. In *IEEE International Conference on Advanced Information Networking and Applications*, pages 467–474, Biopolis, Singapore. IEEE.

Basili, V., Caldiera, G., and Rombach, H. D. (1994). The goal question metric approach. *Encyclopedia of software engineering*, 2:1–10.

Batini, C., Lenzerini, M., and Navathe, S. (1986). A comparative analysis of methodologies for database schema integration. *ACM computing surveys (CSUR)*, 18(4):323–364.

Batty, M., Crooks, A., Hudson Smith, A., Milton, R., Anand, S., Jackson, M., and Morley, J. (2010). Data mash-ups and the future of mapping. Technical report, JISC: Bristol, UK.

Baxter, P. and Susan, J. (2008). Qualitative Case Study Methodology: Study Design and Implementation for Novice Researchers. *The Qualitative Report*, 13(4):544–559.

Bayne, J. (2002). White Paper: An Overview of Threat and Risk Assessment. Technical report.

Begum, B. A., Thakur, R. K., and Patra, P. K. (2010). Security policy integration and conflict reconciliation for data integration across data sharing services in ubiquitous computing environments. In *International Conference on Computer and Communication Technology (ICCCT'10)*, pages 1–6, Allahabad, India. IEEE.

Bhowmick, S., Gruenwald, L., Iwaihara, M., and Chatvichienchai, S. (2006). PRIVATE-IYE: A Framework for Privacy Preserving Data Integration. In *Proceedings of the 22nd International Conference on Data Engineering Workshops (ICDEW'06)*, Washington, DC, USA. IEEE.

Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked data-the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22.

Borders, K. and Prakash, A. (2009). Quantifying information leaks in outbound web traffic. In *30th IEEE Symposium on Security and Privacy*, Oakland,Clifornia, USA.

Boyatzis, R. E. (1998). *Transforming Quaitative Information :Thematic Analysis and Code Development*. SAGE Publications, Inc.

Boyens, C., Krishnan, R., and Padman, R. (2004). On privacy-preserving access to distributed heterogeneous healthcare information. In *Proceedings of the 37th Hawaii International Conference on System Sciences*, pages 1–10, Big Island, Hawaii, USA.

Braghin, C., Cortesi, A., and Focardi, R. (2003). Information leakage detection in boundary ambients. *Electronic Notes in Theoretical Computer Science*, (78):123–143.

Braun, V. and Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 0887(September 2015):77–101.

Brostoff, S. and Sasse, M. A. (2001). Safe and Sound: A Safety-critical Approach to Security. In *New Security Paradigms Workshop*, pages 41–50, Cloudcroft, New Mexico.

Butler, M. (2013). Mastering System Analysis and Design through Abstraction and Refinement. In Manfred Broy, Doron Peled, G. K., editor, *Engineering Dependable Software Systems*, pages 49–78. IOS Press.

Butler, M. J., Leuschel, M., Presti, S. L., and Turner, P. (2004). The Use of Formal Methods in the Analysis of Trust (Position Paper). *Trust Management, Lecture Notes in Computer Science*, 2995:333–339.

Butler, S. A. (2002). Security Design: Why It's Hard To Do Empirical Research.

Caceres, G. H. R. and Teshigawara, Y. (2010). Security guideline tool for home users based on international standards. *Information Management & Computer Security*, 18(2):101–123.

Calì, A., Calvanese, D., Giacomo, G., and Lenzerini, M. (2006). Data integration under integrity constraints. *Advanced Information Systems Engineering*, pages 262–279.

Calvanese, D., Giacomo, G. D., Lenzerini, M., Nardi, D., and Rosati, R. (1998). Description logic framework for information integration. In *The 6th International Conference on the Principles of Knowledge Representation and Reasoning (KR'98)*, number 22469, pages 2–13, Trento, Italy.

Cannon, D. L. (2008). *CISA Certified Information Systems Auditor: Study Guide*. John Wiley & Sons, 2nd edition.

Carey, M. J., Onose, N., and Petropoulos, M. (2012). Data Services. *Communications of the ACM*, 55(6):86–97.

Cattell, R. (2010). Scalable SQL and NoSQL data stores. *ACM SIGMOD Record*, 39(4):12.

Chandra, S. and Khan, R. (2008). Object Oriented Software Security Estimiation Life Cycle-Design Phase Prerpective. *Journal of Software Engineering 2*, 1:39–46.

Chandra, S., Khan, R., and Agrawal, A. (2009). Security Estimation Framework: Design Phase Perspective. In *Sixth International Conference on Information Technology: New Generations*, pages 254–259, Las Vegas, Nevada, USA. IEEE.

Chatzikokolakis, K., Chothia, T., and Guha, A. (2010). Statistical Measurement of Information Leakage. *Tools and Algorithms for the Construction and Analysis of Systems*, 6015:390–404.

Chawathe, S., Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J., and Widom, J. (1994). The TSIMMIS project: Integration of heterogenous information sources. In *Proceedings of the lOth Meeting of the Information Processing Society of Japan*, pages 7–18.

Chess, B. and Arkin, B. (2011). Software Security in Practice. *IEEE Security & Privacy Magazine*, 9(2):89–92.

ching Leung, W. (2001). How to design a questionnaire. *Student BMJ*, 9:187–189.

Chothia, T., Kawamoto, Y., and Novakovic, C. (2013). A tool for estimating information leakage. In *International Conference on Computer Aided Verification*, pages 690–695, Saint Petersburg, Russia. Springer Berlin Heidelberg.

Chung, L. (1993). Dealing with security requirements during the development of information systems. In *Advanced Information Systems Engineering: 5th International Conference, CAiSE '93*, pages 234–251, Paris, France. Springer Berlin Heidelberg.

Clarke, E. M., Wing, J. M., and Others (1996). Formal methods: state of the art and future directions. *ACM Computing Surveys*, 28(4):626–643.

Clifton, C., Kantarciolu, M., Doan, A., Schadow, G., Vaidya, J., Elmagarmid, A., and Suciu, D. (2004). Privacy-preserving data integration and sharing. In *Proceedings of the 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery - DMKD '04*, pages 19–26, Paris, France. ACM Press.

Constante, E., Paci, F., and Zannone, N. (2013). Privacy-aware web service composition and ranking. In *Proceedings - IEEE 20th International Conference on Web Services, ICWS 2013*, pages 131–138, Washington, DC, USA.

Crampton, J. and Huth, M. (2010). Towards an access-control framework for countering insider threats. *Advances in Information Security*, 49:173–195.

Creswell, J. W. (2003). *Research Design Qualitative, Quantitative, and Mixed Methods Approaches*. Sage Publications.

Cronin, P., Ryan, F., and Coughlan, M. (2008). Undertaking a literature review: a step-by-step approach. *British journal of nursing (Mark Allen Publishing)*, 17(1):38–43.

Cruz, I., Gjomemo, R., and Orsini, M. (2008). A Secure Mediator for Integrating Multiple Level Access Control Policies. *Knowledge-Based Intelligent Information and Engineering Systems*, pages 354–362.

CWE (2013). CWE-200: Information Leak (Information Exposure). Available at `http://cwe.mitre.org/data/definitions/200.html` accessed 02-08-2013.

Davis, F. D. (1989). Perceived Usefulness, Perceived Ease Of Use, and User Acceptance. *MIS Quarterly*, 13(3):319–339.

Dayal, U., Castellanos, M., Simitsis, A., and Wilkinson, K. (2009). Data integration flows for business intelligence. In *Proceedings of the 12th International Conference on Extending Database Technology Advances in Database Technology - EDBT '09*, pages 1–11, Saint-Petersburg, Russian Federation. ACM Press.

Deng, M., Wuyts, K., Scandariato, R., and Wouter, B. P. (2011). A privacy threat analysis framework: supporting the elicitation and fulfill...: EBSCOhost. *Requirements Engineering*, 16:3–32.

Devanbu, P. and Stubblebine, S. (2000). Software engineering for security: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering*, pages 225–239, Limerick, Ireland.

Dicelie, J. J., Esenther, A. W., Walsh, T. C., Wong, D. W. H., and Young, M. J. (2001). Patent: Data integration system. Available at `http://google.com/patents/EP1122652A1?cl=zh-cn`.

Doan, A. and Halevy, A. (2005). Semantic integration research in the database community: A brief survey. *AI magazine*, 26(1):83.

Dong, X. L. and Srivastava, D. (2015). Big data integration. *Synthesis Lectures on data Management*, page 198.

Duan, L., Zhang, Y., Chen, S., Zhao, S., Wang, S., Liu, D., Liu, R. P., Cheng, B., and Chen, J. (2016). Automated Policy Combination for Secure Data Sharing in Cross-Organizational Collaborations. *IEEE Access*, 4:3454–3468.

Dumas, J., Sorce, J., and Virzi, R. (1995). Expert Reviews: How Many Experts is Enough? In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 39, pages 228–232, San Diego, California, USA.

European Communities-Commission (1991). Security Evaluation Criteria (ITSEC) (Provisional Harmonised Criteria, Version 1.2, 28 June 1991).

Fereday, J. and Muir-Cochrane, E. (2006). Demonstrating Rigor Using Thematic Analysis : A Hybrid Approach of Inductive and Deductive Coding and Theme Development. *International Journal of Qualitative Methods*, 5:80–92.

Fisler, K., Krishnamurthi, S., Meyerovich, L. A., and Tschantz, M. C. (2005). Verification and Change-Impact Analysis of Access-Control Policies. In *Proceedings of the 27th International Conference on Software Engineering*, pages 196–205, St. Louis,Missouri, USA.

Fléchais, I. (2005). *Designing Secure and Usable Systems*. PhD thesis, University College London.

Fléchais, I., Jirotka, M., and Alghamdi, D. (2013). In the balance in Saudi Arabia: security, privacy and trust. In *Extended Abstracts on Human Factors in Computing Systems CHI '13*, pages 823–828, Paris, France.

Fung, B. C., Trojer, T., Hung, P. C., Xiong, L., Al-Hussaeni, K., and Dssouli, R. (2012). Service-Oriented Architecture for High-Dimensional Private Data Mashup. *IEEE Transactions on Services Computing*, 5(3):373–386.

Futcher, L. and von Solms, R. (2008). Guidelines for secure software development. In *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology*, pages 56–65, Wilderness, South Africa.

Gay, L. R. (1996). *Educational research: competencies for analysis and application*. Prentice-Hall, 10th edition.

Gessiou, E., Vu, Q. H., and Ioannidis, S. (2011). IRILD: An Information Retrieval Based Method for Information Leak Detection. In *the Seventh European Conference on Computer Network Defense*, pages 33–40, Washington, DC, USA. IEEE.

Gill, P., Stewart, K., Treasure, E., and Chadwick, B. (2008). Methods of data collection in qualitative research: interviews and focus groups. *British dental journal*, 204(6):291–295.

Gilliam, D., Wolfe, T., Sherif, J., and Bishop, M. (2003). Software security checklist for the software life cycle. In *Proceedings of the Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises WET ICE '03*, pages 243–248, Linz, Austria. IEEE Comput. Soc.

Goguen, J. A. and Meseguer, J. (1982). Security Policies and Security Models. In *IEEE Symposium on Security and Privacy*, pages 11–20, Oakland, CA, USA.

Golafshani, N. (2003). Understanding reliability and validity in qualitative research. *The Qualitative Report*, 8(4):597–607.

Gollmann, D. (2006). *Computer Security*. John Wiley & Sons, second edition.

Goryczka, S., Xiong, L., and Fung, B. C. M. (2013). m-Privacy for Collaborative Data Publishing. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–13.

Guarda, P. and Zannone, N. (2009). Towards the development of privacy-aware systems. *Information and Software Technology*, 51(2):337–350.

Guo, F. and Fang, Y. (2011). An Ontology-Based Data Integration System with Dynamic Concept Mapping and Plug-In Management. In *International Conference of Information Technology, Computer Engineering and Management Sciences*, pages 324–328, Nanjing, Jiangsu, China. IEEE.

Gupta, A. and Mumick, I. (1995). Maintenance of materialized views: Problems, techniques, and applications. *Data Engineering Bulletin*, 18(2):1–16.

Gurses, S., Troncoso, C., and Diaz, C. (2011). Engineering Privacy by Design. In *the Fourth Conference on Computers, Privacy & Data Protection*, Brussels, Belgium.

Gusmini, A. and Leida, M. (2011). A patent: Data Integration System. Available at `https://www.google.ch/patents/US20130006968`.

Haddad, M., Hacid, M. S., and Laurini, R. (2012). Data Integration in Presence of Authorization Policies. In *IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 92–99, Liverpool, UK. IEEE.

Haerder, T. and Reuter, A. (1983). Principles of transaction-oriented database recovery. *ACM Comput. Surv.*, 15(4):287–317.

Halevy, A., Ashish, N., Bitton, D., Carey, M., Draper, D., Pollock, J., Rosenthal, A., and Sikka, V. (2005). Enterprise information integration: successes, challenges and controversies. In *Proceedings of the International Conference on Management of Data (SIGMOD'05)*, pages 778–787, Baltimore, Maryland, USA.

Halevy, A., Rajaraman, A., and Ordille, J. (2006). Data integration: the teenage years. In *32nd International Conference on Very large data bases VLDB'06*, Seoul, Korea.

Haley, C., Laney, R., Moffett, J. D., and Nuseibeh, B. (2008). Security requirements engineering: A framework for representation and analysis. *IEEE Transactions on Software Engineering*, 34(1):133–153.

Hallerstede, S. (2011). On the purpose of Event-B proof obligations. *Formal Aspects of Computing*, 23(1):133–150.

Harris, D., Khan, L., Paul, R., and Thuraisingham, B. (2007). Standards for secure data sharing across organizations. *Computer Standards & Interfaces*, 29(1):86–96.

Hart, B. (2001). Implementing a Successful Security Assessment Process. Technical report.

Hashizume, K., Rosado, D. G., Fernández-Medina, E., and Fernandez, E. B. (2013). An analysis of security issues for cloud computing. *Journal of Internet Services and Applications*, 4(1):1–13.

He, Q., Annie I. Anton, P. O., and Jones, L. (2006). Ensuring compliance between policies, requirements and software design: A case study. In *Proceedings of the Fourth IEEE International Workshop on Information Assurance (IWIA'06)*, pages 1–14, Washington, DC, USA.

Heikkinen, S., Kinnari, S., and Heikkinen, K. (2009). Security and User Guidelines for the Design of the Future Networked Systems. In *Third International Conference on Digital Society*, pages 13–19, Cancun, Mexico. IEEE.

Hennessy, S., Lauer, G., Zunic, N., Gerber, B., and Nelson, A. (2009). Data-centric security: Integrating data privacy and data security. *IBM Journal of Research and Development*, 53(2):1–12.

Herbert, M. and Thieme, T. (2012). Secure mashup-providing platforms-implementing encrypted wiring. *Current Trends in Web Engineering*, 7059:99–108.

Hoang, T. S., Basin, D., and Abrial, J. R. (2009). Specifying Access Control in Event-B. *Technical Report*, 624.

Holz, H. J., State, C., Bay, E., Applin, A., Joyce, D., Purchase, H., and Reed, C. (2006). Research Methods in Computing: What are they, and how should we teach them? In *The Eleventh Annual Conference on Innovation and Technology in Computer Science Education ITiCSE'06*, pages 96–114, Bologna,Italy.

Horie, D., Kasahara, T., Goto, Y., and Cheng, J. (2009). A New Model of Software Life Cycle Processes for Consistent Design, Development, Management, and Maintenance of Secure Information Systems. In *Eighth IEEE/ACIS International Conference on Computer and Information Science*, pages 897–902, Shanghai, China. IEEE.

Hu, Y. and Yang, J. (2011). A semantic privacy-preserving model for data sharing and integration. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics - WIMS '11*, Sogndal, Norway. ACM Press.

Huang, W., Liu, T., and Zhao, Y. (2008). Honeycomb: A Community-Based System for Distributed Data Integration and Sharing. In *Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies PDCAT 2008*, pages 107–114, Dunedin, Otago, New Zealand. IEEE.

Hughes, G. and Bultan, T. (2008). Automated verification of access control policies using a SAT solver. *International Journal on Software Tools for Technology Transfer*, 10(6):503–520.

Hull, R. and Zhou, G. (1996). A framework for supporting data integration using the materialized and virtual approaches. In *SIGMOD '96 Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, pages 481–492, Montreal, Canada. ACM.

Hung, P. (2005). Towards a privacy access control model for e-healthcare services. In *Third Annual Conference on Privacy, Security and Trust*, pages 12–14, Andrews, New Brunswick, Canada.

ICO (2017). Information security (Principle 7). Available at `https://ico.org.uk/for-organisations/guide-to-data-protection/principle-7-security/` accessed 18-08-2016.

Islam, S. and Falcarin, P. (2011). Measuring security requirements for software security. In *IEEE 10th International Conference on Cybernetic Intelligent Systems (CIS)*, pages 70–75, London,UK. IEEE.

ISO (2014). ISO/IEC27000: Information technology Security techniques Information security management systems Overview and vocabulary. *BSI Standards Publication*.

Jaferian, P., Botta, D., Hawkey, K., and Beznosov, K. (2008). Design guidelines for IT security management tools. In *SOUPS Workshop on Usable IT Security Management (USM)*, pages 1–6, Pitsburgh, PA, USA.

Jawad, M., Serrano-Alvarado, P., and Valduriez, P. (2013). Supporting Data Privacy in P2P Systems. *Security and Privacy Preserving in Social Networks*, pages 1–51.

Joshi, J., Aref, W., Ghafoor, A., and Spafford, E. H. (2001). Security models for web-based applications. *Communications of the ACM*, 44(2):38–44.

Juerjens, J. (2002). Using UMLsec and Goal Trees for Secure Systems Development. In *ACM Symposium on Applied Computing, SAC 2002*, pages 1026–1030, Madrid,Spain.

Jurczyk, P. and Xiong, L. (2008). Towards privacy-preserving integration of distributed heterogeneous data. In *Proceedings of the 2nd PhD workshop on Information and knowledge management*, pages 65–72, Napa Valley, California, USA.

Kasemsan, M. L. K. and Hunngam, N. (2011). Internet Banking Security Guideline Model for Banking in Thailand. *Communications of the IBIMA*, 2011:1–13.

Katic, N., Quirchmay, G., Schiefer, J., Stolba, M., and Tjoa, A. (1998). A prototype model for data warehouse security based on metadata. In *9th International Workshop on Database and Expert Systems Applications (DEXA'98)*, Vienna, Austria. IEEE Computer Society.

Kaufman, S., Rosset, S., and Perlich, C. (2011). Leakage in data mining: Formulation, detection, and avoidance. In *The 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD '11*, pages 556–563, San Diego, California, USA.

Keim, D., Mansmann, F., Schneidewind, J., and Ziegler, H. (2006). Challenges in Visual Data Analysis. In *Tenth International Conference on Information Visualisation (IV'06)*, pages 9–16, London,UK. IEEE.

Khan, M. U. A. and Zulkernine, M. (2009). On Selecting Appropriate Development Processes and Requirements Engineering Methods for Secure Software. In *33rd Annual IEEE International Computer Software and Applications Conference*, pages 353–358, Seattle, WA, USA. IEEE.

Khattak, Z., Sulaiman, S., and Manan, J. (2012). Evaluation of Unified Security, Trust and Privacy Framework (UnifiedSTPF) for Federated Identity and Access Management (FIAM) Mode. *International Journal of Computer Applications*, 54(6):12–19.

Kolovski, V., Hendler, J., and Parsia, B. (2007). Analyzing web access control policies. In *Proceedings of the 16th international conference on World Wide Web - WWW '07*, page 677, Banff, Alberta, Canada.

Kuang, T. and Ibrahim, H. (2009). Security privacy access control for policy integration and conflict reconciliation in health care organizations collaborations. In *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services (iiWAS2009)*, pages 750–754, Kuala Lumpur, Malaysia.

Lamb, P., Power, R., Walker, G., and Compton, M. (2006). Role-based access control for data service integration. In *Proceedings of the 3rd ACM workshop on Secure web services - SWS '06*, pages 3–12, Alexandria, VA,USA. ACM Press.

Langegger, A., Wöß, W., and Blöchl, M. (2008). A semantic web middleware for virtual data integration on the web. *The Semantic Web: Research and Applications*, pages 493–507.

Lawrence, R. (2014). Integration and virtualization of relational SQL and NoSQL systems including MySQL and MongoDB. In *Proceedings - 2014 International Conference on Computational Science and Computational Intelligence, CSCI 2014*, volume 1, pages 285–290, Las Vegas, Nevada, USA.

Le, X. and Wang, D. (2012). Development of a system framework for implementation of an enhanced role-based access control model to support collaborative processes. In *Proceedings of the 3rd USENIX conference on Health Security and Privacy*, Bellevue, WA, USA. USENIX Association.

Leavitt, N. (2010). Will NoSQL Databases Live Up to Their Promise? *Computer*, 43(2):12–14.

Lenzerini, M. (2002). Data integration: A theoretical perspective. In *Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 233–246, Madison, Wisconsin, USA.

Leuschel, M. and Butler, M. (2003). The ProB Animator and Model Checker for B - A Tool Description. *International Symposium of Formal Methods Europe*, 2805:855–874.

Li, F., Luo, B., Liu, P., Lee, D., and Chu, C.-H. (2013). Enforcing Secure and Privacy-Preserving Information Brokering in Distributed Information Sharing. *IEEE Transactions on Information Forensics and Security*, 8(6):888–900.

Li, M., Wang, H., and Plank, A. (2009). Privacy-aware access control with generalization boundaries. In *Proceedings of the 32nd Australasian Computer Science Conference (ACSC 2009)*, number Acsc, pages 105–112, Wellington, New Zealand.

Likert, R. (1932). *A technique for the measurement of attitudes*.

Lister, R. (2005). Mixed Methods: Positivists are from Mars, Constructivists are from Venus. *SIGCSE Bull*, 37(4):18–19.

Lorenzo, G. D., Hacid, H., Paik, H., and Benatallah, B. (2009). Data integration in mashups. *ACM Sigmod Record*, 38(1):59–66.

Loshin, D. (2010). White paper on: Data Integration Alternatives Managing Value and Quality. *Pitney Bowes*.

Manan, J. A., Mubarak, M., and Isa, M. (2011). Security, Trust and PrivacyA New Direction for Pervasive Computing. In *Proceedings of the 15th WSEAS international conference on Computers*, pages 56–60, Stevens Point, Wisconsin, USA.

Marshall, B., Cardon, P., Poddar, A., and Fontenot, R. (2013). Does Sample Size Matter in Qualitative Research ?: a Review of Qualitative Interviews in IS Research. *Journal of Computer Information Systems*, 54(1):11–22.

Mathison, S. (1988). Why Triangulate? *Educational Researcher*, 17(2):13–17.

Maude, F. (2012). Open Data White Paper-Unleashing the potential. *Cabinet Office (London)*, pages 1–51.

Maxion, R. A., Longstaff, T. A., and McHugh, J. (2010). Why is there no science in cyber science?: a panel discussion at NSPW 2010. In *Proceedings of the 2010 workshop on New security paradigms*, pages 1–6, Concord, Massachusetts, USA.

Mccallister, E., Grance, T., and Scarfone, K. (2010). Guide to Protecting the Confidentiality of Personally Identifiable Information (PII) Recommendations of the National Institute of Standards and Technology. *NIST Special Publication (800-122)*, page 59.

McConnell, S. (2004). *Code Complete*. Microsoft Press.

McGraw, G. (2004). Software security. *IEEE Security & Privacy Magazine*, pages 80–83.

McGraw, G. (2013). Cyber War is Inevitable (Unless We Build Security In). *Journal of Strategic Studies*, 36(1):109–119.

Meier, J., Mackman, A., Dunner, M., Vasireddy, S., Escamilla, R., and Murukan, A. (2003). Threat Modeling- Improving Web Application Security: Threats and Countermeasures. Available at `https://msdn.microsoft.com/en-us/library/ff648644.aspx?f=255&MSPPError=-2147217396` accessed 04-07-2016.

Meingast, M., Roosta, T., and Sastry, S. (2006). Security and privacy issues with health care information technology. In *Proceedings of the 28th IEEE Annual International Conference of Engineering in Medicine and Biology Society.*, volume 1, pages 5453–5458, New York, New York, USA.

Mir, I. A. and Quadri, S. (2012). Analysis and Evaluating Security of Component-Based Software Development: A Security Metrics Framework. *International Journal of Computer Network and Information Security*, 4(11):21–31.

Mohammed, N. and Fung, B. (2010). Centralized and distributed anonymization for high-dimensional healthcare data. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 4(4):18–33.

Mohammed, N., Fung, B. C. M., and Debbabi, M. (2011). Anonymity meets game theory: secure data integration with malicious participants. *the International Journal on Very Large Data Bases*, 20(4):567–588.

Morton, A. and Sasse, M. (2012). Privacy is a process, not a PET: a theory for effective privacy practice. In *Proceedings of the 2012 workshop on new security paradigms NSPW'12*, pages 87–104, Bertinoro, Italy.

Mouratidis, H., Giorgini, P., and Manson, G. (2005). When security meets software engineering: a case of modelling secure information systems. *Information Systems*, 30(8):609–629.

Myagmar, S., Lee, A., and Yurcik, W. (2005). Threat modeling as a basis for security requirements. In *Symposium on Requirements Engineering for Information Security (SREIS)*.

Nachouki, G. and Quafafou, M. (2011). MashUp web data sources and services based on semantic queries. *Information Systems*, 36(2):151–173.

Newcombe, C., Rath, T., Zhang, F., Munteanu, B., Brooker, M., and Deardeuff, M. (2015). How Amazon web services uses formal methods. *Communications of the ACM*, 58(4):66–73.

Nielsen, J. and Mack, R. (1994). *Heuristic Evaluation*. John Wiley & Sons.

Noor, T. H., Sheng, Q. Z., Zeadally, S., and Yu, J. (2013). Trust management of services in cloud environments: Obstacles and solutions. *ACM Computing Surveys*, 46(1):12–30.

Noy, N. (2004). Semantic integration: a survey of ontology-based approaches. *SIGMOD record*, 33(4):65–70.

Nurse, J. R. C., Creese, S., Goldsmith, M., and Lamberts, K. (2011). Guidelines for usable cybersecurity: Past and present. In *Third International Workshop on Cyberspace Safety and Security (CSS)*, pages 21–26, Milan, Italy. IEEE.

Paci, F., Fernandez-Gago, C., and Moyano, F. (2013). Detecting insider threats: A trust-aware framework. In *Proceedings of the International Conference on Availability, Reliability and Security, ARES 2013*, pages 121–130, University of Regensburg, Germany.

Palanimalai, S. and Paramasivam, I. (2015). An Enterprise Oriented View on the Cloud Integration Approaches Hybrid Cloud and Big Data. *Procedia Computer Science*, 50:163–168.

Pasierb, K., Kajdanowicz, T., and Kazienko, P. (2011). Privacy-preserving data mining, sharing and publishing. *Journal of Medical Informatics & Technologies*, 18:70–76.

Pearson, S. (2009). Taking Account of Privacy when Designing Cloud Computing Services. In *ICSE Workshop on Software Engineering Challenges of Cloud Computing, 2009. CLOUD '09*, pages 44–52, Vancouver, BC.

Pistoia, M., Fink, S. J., Flynn, R. J., and Yahav, E. (2007). When Role Models Have Flaws: Static Validation of Enterprise Security Policies Introduction: RBAC Systems. In *29th International Conference on Software Engineering*, Minneapolis, MN, USA.

Pon, R. and Critchlow, T. (2005). Performance-oriented privacy-preserving data integration. *Data Integration in the Life Sciences*, pages 240–256.

Prakash, V. and Darbari, M. (2012). A Review on Security Issues in Distributed Systems. *International Journal of Scientific & Engineering*, 3(9):1–5.

Ramli, C. D. P. K., Nielson, H. R., and Nielson, F. (2013). XACML 3.0 in Answer Set Programming. *Logic-Based Program Synthesis and Transformation*, 7844:89–105.

Ray, S. S., Bandyopadhyay, S., and Pal, S. K. (2009). Combining multisource information through functional-annotation-based weighting: gene function prediction in yeast. *IEEE transactions on bio-medical engineering*, 56(2):229–36.

Reeve, A. (2013). Cloud-Based Data Integration Adds Concerns about Latency and Security. Available `http://data-informed.com/cloud-based-data-integration-adds-concerns-about-latency-and-security/`, accessed 04-02-2014.

Ristenpart, T., Tromer, E., Shacham, H., and Savage, S. (2009). Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In *Proceedings of the 16th ACM conference on Computer and communications security*, Chicago, Illinois, USA.

Ross, R., Oren, J. C., and Mcevilley, M. (2014). Systems security engineering an integrated approach to building trustworthy resilient systems. Technical report.

Runeson, P. and Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2):131–164.

Russom, P. (2008). Data Integration Architecture: What It Does, Where It's Going, and Why You Should Care. Available at `http://tdwi.org/articles/2008/05/27/data-integration-architecture-what-it-does-where-its-going-and-why-you-should-care.aspx` accessed 26-02-2017.

Saeed, M. Y., Tahir, A., Mughal, S., and Khan, M. N. A. (2014). Insight into Security Challenges for Cloud Databases and Data Protection Techniques for Building Trust in Cloud Computing. *Journal of Basic and Applied Scientific Research*, 4(1):54–59.

Schaar, P. (2010). Privacy by Design. *Identity in the Information Society - Special Issue*, 3(2):267–274.

Seaman, C. (1999). Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, 25(4):557–572.

Shafiq, B., Joshi, J., Bertino, E., and Ghafoor, A. (2005). Secure interoperation in a multidomain environment employing RBAC policies. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1557–1577.

Sharma, V. and Dave, M. (2012). SQL and NoSQL Databases. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(8):2277–128.

Shaw, M. (2002). What makes good research in software engineering? *International Journal on Software Tools for Technology Transfer*, 4:1–7.

Siponen, M. T. and Oinas-kukkonen, H. (2007). A Review of Information Security Issues and Respective Contributions. *The Data Base for Advances in Information Systems*, 38(1):60–80.

Stake, R. E. (1995). *The Art of Case Study Research*. Sage Publications.

Sun, D., Chang, G., Sun, L., and Wang, X. (2011). Surveying and Analyzing Security, Privacy and Trust Issues in Cloud Computing Environments. *Procedia Engineering*, 15:2852–2856.

Sweeney, L. (2002). Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):571–588.

Takabi, H., Joshi, J. B. D., and Ahn, G. J. (2010). Security and privacy challenges in cloud computing environments. *IEEE Security & Privacy Magazine*, (December):24–31.

Taylor, G. R. (2005). *Integrating Quantitative and Qualitative Methods in Research*. University Press of America, 2nd revise edition.

Tipton, H. F. and Nozaki, M. K. (2007). *Information Security Management Handbook*. CRC Press, sixth edition.

Torr, P. (2005). Demystifying the threat modeling process. *IEEE Security & Privacy Magazine*, 3:66–70.

Treglia, J. V. and Park, J. S. (2009). Towards trusted intelligence information sharing. In *Proceedings of the ACM SIGKDD Workshop on CyberSecurity and Intelligence Informatics - CSI-KDD '09*, pages 45–52, Paris, France. ACM Press.

Trumbull, M. (2005). Qualitative Reseach Methods. In *Integrating Quantitative and Qualitative Methods in Research*. University Press of America.

Turkmen, F., Den Hartog, J., Ranise, S., and Zannone, N. (2015). Analysis of XACML policies with SMT. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9036:115–134.

UK Parliament (1998). Data Protection Act 1998. Available at `http://www.legislation.gov.uk/ukpga/1998/29/contents`.

Ullman, J. D., Garcia-Molina, H., and Widom, J. (2001). Infromation Integration. In *Database Systems: The Complete Book*. Prentice Hall.

van den Braak, S. W., Choenni, S., Meijer, R., and Zuiderwijk, A. (2012). Trusted third parties for secure and privacy-preserving data integration and sharing in the public sector. In *Proceedings of the 13th Annual International Conference on Digital Government Research - dg.o '12*, pages 135 –144, College Park, MD, USA. ACM Press.

Viega, J., Kohno, T., and Potter, B. (2001). Trust (and mistrust) in secure applications. *Communications of the ACM*, 44(2):31–36.

von Solms, B. and von Solms, R. (2004). The 10 deadly sins of information security management. *Computers and Security*, 23(5):371–376.

von Solms, R. and van Niekerk, J. (2013). From information security to cyber security. *Computers and Security*, 38:97–102.

Wache, H., Vogele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., and Hubner, S. (2001). Ontology-Based Integration of Information A Survey of Existing Approaches. In *IJCAI-01 workshop: ontologies and information sharing*, pages 108–117, Seattle ,WA, USA.

Walton, G., Longstaff, T., and Linger, R. (2009). Computational evaluation of software security attributes. In *Proceedings of the 42nd Hawaii International Conference on System Sciences*, pages 1–10, Grand Wailea, Maui, HI, USA.

Wang, C. and Wulf, W. (1997). Towards a framework for security measurement. In *20th National Information Systems Security Conference*, Baltimore, MD,USA.

Wang, K., Fung, B., and Dong, G. (2005). Integrating private databases for data analysis. *Intelligence and Security Informatics*, pages 171–182.

Wang, Q. and Jin, H. (2011). Data leakage mitigation for discretionary access control in collaboration clouds. In *Proceedings of the 16th ACM symposium on Access control models and technologies - SACMAT '11*, pages 103–112, Innsbruck, Austria.

Watson, D. (2007). Web Application Attacks. *Network Security*, (October):10–14.

Westin, A. F. (1970). *Privacy and Freedom*. The Bodley Head Ltd.

Whang, S. and Garcia-Molina, H. (2012). A model for quantifying information leakage. In *Secure Data Management: 9th VLDB Workshop, SDM 2012*, pages 25–44. Springer Berlin Heidelberg.

Whittaker, J. and Howard, M. (2004). Building more secure software with improved development processes. *IEEE Security & Privacy Magazine*, pages 63–65.

Wiederhold, G. (1993). Intelligent integration of information. *ACM SIGMOD Record*, pages 434–437.

Wing, J. M. (1998). A Symbiotic Relationship Between Formal Methods and Security. In *Proceedings Conference on Computer Security Dependability and Assurance From Needs to Solutions*, pages 26–38, Washington, DC, USA.

Wood, C. C. (2005). *Information Security Policies Made Easy, Version 10.*

Xiong, L., Chitti, S., and Liu, L. (2007). Preserving data privacy in outsourcing data aggregation services. *ACM Transactions on Internet Technology*, 7(3):28.

Yau, S. S. and Yin, Y. (2008). A Privacy Preserving Repository for Data Integration across Data Sharing Services. *IEEE Transactions on Services Computing*, 1(3):130–140.

Yin, R. k. (1984). *Case Study Resaerch Design and Methods.* Sage Publications.

Yiu, C. (2012). The Big Data Opportunity. *Policy Exchange*, pages 1–34.

Youssef, A. and Alageel, M. (2012). A Framework for Secure Cloud Computing. *International Journal of Computer Science*, 9(4):487–500.

Zanioli, M., Ferrara, P., and Cortesi, A. (2012). SAILS: static analysis of information leakage with Sample. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing SAC '12*, New York, NY, USA.

Zhang, D. Y., Zeng, Y., Wang, L., Li, H., and Geng, Y. (2011). Modeling and evaluating information leakage caused by inferences in supply chains. *Computers in Industry*, 62(3):351–363.

Zhou, G., Hull, R., King, R., and Franchitti, J. (1995). Data Integration and Warehousing Using H2O. *Data Engineering Bulletin*, 18(2):30–40.

Zimmermann, H. (1980). OSI Reference Model-The ISO Model of Architecture for Open Systems Interconnection. *IEEE Transactions on Communications*, 28(4):425–432.