The Pennsylvania State University
The Graduate School
College of Information Sciences and Technology

DESIGNING AND SECURING DISCOVERY SERVICES FOR EPCGLOBAL NETWORK IN SUPPLY CHAINS

A Dissertation in
Information Sciences and Technology

by
Bing Liu

© 2015 Bing Liu

Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

August 2015
The dissertation of Bing Liu was reviewed and approved* by the following:

Chao-Hsien Chu  
Professor of Information Sciences and Technology  
Dissertation Advisor  
Chair of Committee

Dinghao Wu  
Assistant Professor of Information Sciences and Technology

Jens Grossklags  
Assistant Professor of Information Sciences and Technology

Akhil Kumar  
Professor of Information Systems  
Department of Supply Chain and Information Systems

Carleen Maitland  
Associate Professor of Information Sciences and Technology  
Interim Associate Dean for Undergraduate and Graduate Studies

* Signatures are on file in the Graduate School
ABSTRACT

Internet of Things (IoT) is an emerging, global scale, Internet-based information service architecture enabled by the Radio Frequency Identification (RFID) and sensors technologies for information sharing and corresponding service discoveries. The EPCglobal network is considered as a sub-project under the IoT umbrella, which was created to develop a universal identification system and an open architecture to provide interoperability in a complex multi-vendor scenario. This universal identification system is based on the allocation of a unique ID—Electronic Product Code (EPC)—to every item. The EPC key is a globally unique identifier that is carried by the RFID tag. As a result, the EPCglobal network is an architecture proposed for enabling the sharing of information about the individually identifiable objects among organizations. As such, the life history of each individual object is shared and linked to an object through the unique EPC key. Since a huge volume of data is involved, a suitable service oriented architecture (SOA), which is called EPCglobal Discovery Service (EPCDS), is needed for locating both the key and additional information associated with it. Although several EPCDS models have been introduced by existing studies, most of them are in the very early development stages and cannot fulfill the design requirements, especially in advanced services, network performance, and security aspects.

This dissertation introduces an advanced EPCDS model, which centers around three main tasks. First, we discuss the architecture of EPCglobal network and the needs for advanced discovery services. We examine the detailed EPCDS design requirements that are classified into three categories, functionality and service, network performance, and security and privacy. We select several typical EPCDS models that are proposed in other studies for detailed discussion, and then summarize and analyze current EPCDS models against the design requirements. The
conclusion points out how to take advantage of current EPCDS designs to eliminate the gaps for further exploration.

Next, according to the design requirements and the assessment, we propose an advanced EPCDS design. The enhanced EPCDS network is based on a peer-to-peer infrastructure and a consistent hashing method, adopted for the EPC key assignment function. It enhances the functionalities and services of the system, especially core functions such as publishing and lookup services. Our model can fulfill all the design requirements. In particular, compared to other models, the design is a highly distributed network, which is able to provide both basic and advanced services, as well as superior networking capability, such as reliability, scalability, extensibility, etc. It could mitigate some of the aforementioned shortcomings without completely abandoning the established standard.

Finally, we investigate security threats and vulnerabilities for the RFID-enabled EPCglobal network in detail. We categorize the security issues into several categories, and discuss how to resolve them. To address the challenges posed by complex and dynamic EPC objects and events, we propose a relationship-based federated access control model for EPC Discovery Service, which can be tightly integrated with the proposed EPCDS infrastructure, to fulfill all the security and privacy design requirements. To the best of our knowledge, this is the first study that identifies and examines in detail the causes and impacts of key publishing security threats and interdependent security issues in an EPCDS context.
# TABLE OF CONTENTS

LIST OF FIGURES .......................................................................................................................... vii
LIST OF TABLES ............................................................................................................................... viii
ACKNOWLEDGEMENTS ..................................................................................................................... ix

Chapter 1: Introduction ..................................................................................................................... 1
  1.1. Problem Statement .................................................................................................................... 1
  1.2. Background and Motivation ..................................................................................................... 4
    1.2.1. Internet of Things and RFID Architecture ............................................................................ 4
    1.2.2. EPCglobal Network Architecture ........................................................................................ 6
    1.2.3. Motivation ............................................................................................................................. 11
  1.3. Research Framework and Contributions ................................................................................ 15
    1.3.1. Research Framework ............................................................................................................ 15
    1.3.2. Contributions of this work .................................................................................................. 17
  1.4. Organization of the Dissertation .............................................................................................. 18

Chapter 2: Background ..................................................................................................................... 20
  2.1. RFID and EPCglobal Network ................................................................................................ 20
  2.2. EPCDS Models ......................................................................................................................... 20
  2.3. Peer-to-Peer Computing .......................................................................................................... 21
  2.4. Access Control Methods ......................................................................................................... 22

Chapter 3: Review of Current EPCDS Models ................................................................................. 25
  3.1. EPCDS Design Requirements .................................................................................................. 26
  3.2. Selected EPCDS Models ......................................................................................................... 37
    3.2.1. BRIDGE Project .................................................................................................................. 38
    3.2.2. Aggregate Discovery Service (ADS) ..................................................................................... 45
    3.2.3. Service Location Protocol (SLP) .......................................................................................... 47
    3.2.4. Service Lookup Service (SLS) ............................................................................................. 49
    3.2.5. P2P Approach Based Models ............................................................................................... 50
  3.3. Summary of Current EPCDS Models Against Design Requirements ...................................... 54
    3.3.1. Functionality and Network Performance ............................................................................ 54
    3.3.2. Security and Privacy .......................................................................................................... 57
  3.4. Discussion and Implications .................................................................................................... 61
    3.4.1. Research Direction .............................................................................................................. 61
    3.5. Conclusion ............................................................................................................................... 62

Chapter 4: Peer-to-Peer EPC Discovery Service Network Design .................................................... 64
  4.1. Introduction ............................................................................................................................... 64
  4.2. Proposed EPC Discovery Service Design ................................................................................ 65
    4.2.1. Network Topology and Infrastructure .................................................................................. 67
    4.2.2. Key Assignment Function .................................................................................................. 69
    4.2.3. Publishing Service .............................................................................................................. 72
    4.2.4. Lookup Service .................................................................................................................... 76
    4.2.5. Advanced Services ............................................................................................................. 80
  4.3. Analysis and Evaluation .......................................................................................................... 82
LIST OF FIGURES

Figure 1.1 RFID Sub-System Architecture ......................................................................................... 5
Figure 1.2 The EPCglobal Network Architecture .................................................................................. 7
Figure 1.3 EPC Discovery Service Framework ...................................................................................... 8
Figure 1.4 ONS Looking Up Process ..................................................................................................... 10
Figure 1.5 Research Framework ........................................................................................................... 16
Figure 1.6 Research Problems and Solutions ....................................................................................... 18

Figure 3.1 Example of Interdepend Security Problem ........................................................................... 36
Figure 3.2 Directory Services Model .................................................................................................... 42
Figure 3.3 Query Relay Model ............................................................................................................. 44
Figure 3.4 Aggregate Discovery Service Model ..................................................................................... 46
Figure 3.5 Affilias Extensible Supply Chain Discovery Service ............................................................. 48
Figure 3.6 Service Lookup Service Query Procedure ........................................................................... 50
Figure 3.7 DHT-P2P Architecture ...................................................................................................... 52

Figure 4.1 Proposed EPCDS framework ............................................................................................... 66
Figure 4.2 EPCDS Network Topology and Infrastructure ..................................................................... 68
Figure 4.3 An Example of 3 Digits CPN Key Assignment ...................................................................... 71
Figure 4.4 Publishing Service Architecture ........................................................................................ 75
Figure 4.5 Lookup Service Architecture .............................................................................................. 77
Figure 4.6 Peer to Peer based Aggregating Discovery Services Model ................................................. 79
Figure 4.7 Advanced Service Module Architecture .............................................................................. 81
Figure 4.8 Implemented Prototype Architecture ................................................................................ 84
Figure 4.9 Time complexity of Publishing and Lookup Services ........................................................ 91
Figure 4.10 Path Length of Lookup Service ........................................................................................ 91

Figure 5.1 EPC Discovery Service Framework ..................................................................................... 95
Figure 5.2 An Integrated DS and Access Control Framework .............................................................. 107
Figure 5.3 PKI Model based CA Architecture ...................................................................................... 110
Figure 5.4 EPCDS Federated Authentication Process ......................................................................... 115
Figure 5.5 Sample SAML <AuthnRequest> ....................................................................................... 116
Figure 5.6 Sample SAML <AuthnResponse> ..................................................................................... 117
Figure 5.7 Subjects and Object Ontology Models .............................................................................. 120
Figure 5.8 The Result Aggregation Process ......................................................................................... 124
Figure 5.9 The Security Rule Aggregation Process .............................................................................. 126
Figure 5.10 The Decision Aggregation Process ..................................................................................... 127
Figure 5.11 An Example of Source Data Ontology ............................................................................... 130
Figure 5.12 Example of Relational Data ............................................................................................. 136
Figure 5.13 A Real Case Example ....................................................................................................... 137
Figure 5.14 Average Time Based on Rn and ISn Variables .................................................................. 144
Figure 5.15 Average Total Time Trend Based on Rn and ISn Variables ............................................... 145
Figure 5.16 TdL and TdG Comparison Based on Rn and ISn Variables ............................................... 146
LIST OF TABLES

Table 1.1 Differences between EPCONS and DS ................................................................. 12
Table 3.1 Existing Studies for EPCDS Design ................................................................. 56
Table 3.2 Existing Studies for EPCglobal Security ......................................................... 60
Table 4.1 Example of EPCDS Item Table ..................................................................... 73
Table 4.2 Example of EPCDS DS Table ................................................................. 74
Table 4.3 Experiment Parameters .............................................................................. 86
Table 4.4 Key Distribution in Each Node ................................................................. 89
Table 4.5 Load Balance Chi-Square Test Result ......................................................... 90
Table 5.1 Example EPCIS Database ........................................................................ 96
Table 5.2 Taxonomy of Vulnerabilities in EPC-enabled System .................................. 97
Table 5.3 Threats Taxonomy of EPCglobal System .................................................... 102
Table 5.4 Countermeasures for RFID Sub-system Level ........................................... 104
Table 5.5 An Example of Item Table ........................................................................ 108
Table 5.6 Illustration of the Immediate Upstream and Downstream Relationship ...... 135
Table 5.7 EPCDS Input and Output of Case 1 ............................................................. 137
Table 5.8 The Input and Output of Each Module of Case 2 ......................................... 138
Table 5.9 The Input and Output of Each Module of Case 3 ......................................... 139
Table 5.10 Final Results Comparison of Three Cases .................................................. 139
Table 5.11 Experiment Result Matrix .......................................................................... 143
ACKNOWLEDGEMENTS

At the end of my doctorate study, I would like to show my deepest gratitude to all the people who have ever helped and instructed me.

Especially, I would like to express my special appreciation and thanks to my PhD advisor, Professor Chao-Hsien Chu, for supporting me during my PhD journey. He has been supportive and patient for guiding me towards becoming an independent scholar. I would like to thank him for encouraging my research and for allowing me to grow as a research scientist. His advice on both research as well as on my career have been invaluable.

I would also like to thank my committee members, Professor Dinghao Wu, Professor Jens Grossklags, Professor Akhil Kumar and Professor Jack Hayya for serving as my committee members. I want to thank them for their informative comments and suggestions.

I also want to thank my colleagues of the Smart Sensing Research Laboratory with whom I have worked with during the course of my graduate studies and researches. Their valuable suggestions have helped me to understand my research area better.

Finally and most importantly, I want to say a heartfelt thank you to my parents for always believing in me and encouraging me to follow my dreams. My parents has been a constant source of love, concern, support and strength in whatever way they could during my life.
Chapter 1: Introduction

1.1. Problem Statement

The convergence of the Internet, wireless communications, sensor networks, information processing, and integration technologies has given birth to a new field of study and applications called the Internet of Things (IoT) (Gershenfeld et al., 2004). This paradigm has been widely adopted and used in our daily life, such as in healthcare, transportation, energy consumption, entertainment, the environment, and homes.

The IoT describes a collective Internet-based information service architecture that is deployed globally. In the vision of this concept, all the items are equipped with a Radio Frequency Identification (RFID) or similar function tag to increase transparency and facilitate the information exchange of the products in the global supply networks. Besides the initial application scope, IoT ultimately aims to serve as a backbone for ubiquitous computing, enabling smart environments to easily recognize and identify objects, and retrieve information from the Internet.

As the IoT is growing rapidly, a huge amount of data of real-world objects and events will be available through the association of RFID technologies. RFID is one of the fundamental technologies for automatic identification, data capture, and object tracking. It has been quickly emerging as a popular technology that can be broadly applied in many places. It is able to facilitate the IoT in exchanging information and the management of merchandise and resources in the supply network. As a result, the EPCglobal network was designed for the purpose of standardizing and diffusing the use of RFID and sensor technologies, and is considered as a sub-
project under the IoT. In this network, the Electronic Product Code (EPC) is a unique identifier (ID) embedded into each RFID tag (EPCglobal, 2010b) that is attached to every item. It aims at developing a universal identification system and an open architecture to provide interoperability in a complex multi-party environment. The greatest promise is to enable the seamless sharing of data and item-level visibility for all the participants across the supply network, and to eventually achieve the blueprint of the IoT that every participant could easily share and access valuable business data in a secure manner.

On the other hand, the EPCglobal network is also constrained by several limitations, for instance, high cost, lack of standardization, security and privacy issues, information sharing issues, the voluminous data (Big Data) issues, and the scalability issues (Michael and McCathie, 2005; Juels, 2006). In order to take the advantage of the EPCglobal network, information from different organizations should and must be shared for facilitating each business to achieve a smart and highly networked environment. EPCglobal is a global scaled network involving vast volumes of data generated by the RFID and sensor devices, and stored in widely distributed and heterogeneous databases. In order to respond to the worldwide query demands from business partners and individual consumers, a discovery service system is needed to provide the relevant services. While EPCglobal is improving people’s daily life, problems such as information security and large-scale data management are critical and should be studied before the EPCglobal systems are deployed. A highly efficient and scalable model for discovery services is the foundation for this ubiquitous computing.

Although the EPCglobal architecture defines a functional component called EPC Discovery Services (EPCDS), no detailed specification has been developed, and many problems (e.g., standardization, high cost, limited services provided, and inefficiency) remain to be solved. In addition, a systematic and explicit design requirement framework is missing. Along with the
information sharing benefits generated by the network, there are serious security challenges due to the lack of appropriate protections. Several discovery service models have been proposed to meet different functionality and services requirements (to be reviewed in details in Chapter 3); most models, however, are designed as intermediate brokers for simple lookup services, which cannot fulfill clients’ expectations and needs. Also, they can hardly convince us that they have good network performance, and are able to handle global-scale distributed data. Thus, a comprehensive and detailed discussion and assessment of each model is needed for directing future design and implementation. Furthermore, a new EPCDS design is expected to provide more of the favorite functions and services, improve networking performance, and make the EPCglobal network more secure.

Finally, there is the issue of security and privacy, and in particular, the data within the EPCglobal network, such as information about price, location, volume and flow of products, and relationships between trading partners, etc., is always commercially sensitive and complicated. In the case of missing the proper access control mechanism, this information can be accessed by any parties that include the hostile entities and the competitor organizations. Because of such concerns, certain security countermeasures, such as key publishing control and federated access control, should be integrated with the EPCglobal network to allow network participants to obtain the power of controlling their own data, since the information is decentralized across multiple organizations (Vanalstyn et al., 1995). While the security and privacy of each organization is guaranteed, they can share their information with more confidence, and thus obtain further benefits.
1.2. Background and Motivation

1.2.1. Internet of Things and RFID Architecture

RFID is a communication and identification technique known at least since the Second World War from friend-or-foe identification systems of military airplanes (Rieback et al., 2006). However, it has been used in more and more other domains since then, e.g., animal or human identification, anti-counterfeiting, and access control payment to global supply chains (Garfinkel and Rosenberg, 2005). Currently, it has been considered as an important underlying technology for smart and ubiquitous computing. The term IoT refers to the networked services that speak about things, rather than services that reside inside the objects themselves (Liu and Albitz, 2006). People are interested in the data and information associated with these things. This valuable information will be available on the internet because of the IoT. It has been noted that the cost of establishing a real IoT is not affordable under current situations; in order to make this dream come true, people have put their hope in a simple chip called the RFID tag, which is mostly externally powered and communicates via radio waves issued by RFID readers (EPCglobal, 2007a). In the recent years the RFID technology has become cheaper and cheaper, and the massive deployment of these tags will most likely be achieved in the near future. They work with the RFID reader, which interrogates all the tags that are close to it via radio waves of a specific frequency. The reader also provides energy to the passive tags. The reader and tags follow an anti-collision protocol to initiate the probe, and then the tags return the data to the reader. Because of the capability of the passive tags, they can only do simple operations and store very limited data, e.g., EPC keys. Besides carrying the identifier data, most of other processes are done by the back-end system and the application layer. It enables these data to be accessed via the Internet, e.g., via different companies’ Web services or EPCDS’s query interfaces.
RFID systems can be very complex and may be integrated with many other applications. Traditionally, a simple RFID sub-system includes hardware and software, as shown in Figure 1.1. The hardware is composed of tags, readers, and a back-end system. The software system contains a user-interface, middleware, and database system. An RFID enabled network, such as an EPCglobal network, is constructed by many RFID sub-systems that are deployed in different stages of supply chains.

![Figure 1.1 RFID Sub-System Architecture](image)

An RFID tag is a small electronic device that consists of a small chip and an antenna to carry and transmit useful information, such as the identity data. The major characteristics of an RFID tag include identifier format, power source, operating frequencies, functionality, and form factor (Karygiannis et al., 2007). According to these characteristics, the RFID tag can be divided into several categories, which greatly differ in many aspects, e.g., the cost, size, performance, and security features. Because of such variety of the RFID tag, it will always need to be customized in order to meet the requirement of certain applications.

Generally, the RFID reader can read the RFID tag, and in some cases write into the tag. It is obvious that the communication cannot happen between different types of RFID tags and RFID readers, unless they are under the same communication protocol. Basically, the RFID tag and reader that are produced by different vendors cannot connect with each other. The prominent
international RFID standard is the EPCglobal Class-1 Generation-2 standard, which is essentially equivalent to the ISO/IEC 18000-6C standard.

After the RFID reader extracts the data from the RFID tag, it sends the data to the back-end system. The back-end system is responsible for the storage and maintenance of that information. It contains two major components: the middleware and the analytic system. RFID middleware is responsible for preparing the collected data for the analytic systems. For instance, the middleware filters duplicate, incomplete, and erroneous information that it receives from readers. The middleware can immediately transfer the filtered data to the analytic systems, which aggregate and store them for later retrieval. It also can be used to monitor and manage readers. For example, system administrators use middleware to adjust the power output and duty cycle to reduce the number of transaction errors. Analytic systems directly support business processes. They are mainly composed of Web servers and databases, and support each customized business logic. In the EPCglobal context we normally consider the backend system as a database.

1.2.2. EPCglobal Network Architecture

EPCglobal is an activity of the global not-for-profit standards organization, which supports the global adoption of the electronic product code (EPC) and related industry driven standards to enable accurate, immediate, and cost-effective visibility of information throughout the supply chain. The EPCglobal Architecture Framework is a collection of interrelated hardware, software, and data standards, together with shared network services that are operated by EPCglobal, its delegates, and others, all in service of this common goal (EPCglobal, 2010a).

Traditionally, EPCglobal network is composed five major components: EPC tag, RFID reader, Application Level Events (ALE), EPC Information Services (EPCIS), and EPC Object Name Service (EPCONS) (see Figure 1.2.)
The working process is the following: the EPC tag will capture the events of the physical object exchange among different organizations. The captured raw data will then be processed and delivered to EPCIS by ALE, which was originally called Savant. ALE is essentially a middleware performing the counting, logging, accumulation, and filtering operations on the data. EPCIS contains two major components—the EPCIS repository and EPCIS query interface. The filtered data will be stored in the EPCIS repository. When the users in the EPCglobal network request information about a specific object, a query will be initiated via the EPCIS query interface. The interface will then interact with the upper layer applications, such as the EPCONS and EPCDS, to extract the target resources.

The EPC is a unique identification number that is used to retrieve information associated with the object in the supply chain. The EPC Tag Data Standard (TDS) (EPCglobal, 2010b) defines several encoding URI syntax schema, such as serialized global trade item number (SGTIN), Serial Shipping Container Code (SSCC), and General Identifier (GID). EPC tags are
potentially the most important class of RFID tags, and constitute the physical embodiment of the EPC to be attached or integrated into supply chain pallets and transporting cases, and possibly to all applicable manufactured single goods in the future. An example of an SGTIN-96 encoded schema—the most popular standard, the serialized global trade identification number (SGTIN)—is given in Figure 1.3.

In this SGTIN-96 variant, the EPC includes a header to denote its EPC identity type, e.g., here is SGTIN-96, a filter value for fast logistic decisions, a partition value that indicates the boundary of the next two fields, and a Company Prefix number (CPN) that is a unique identifier of the item’s manufacturer. Furthermore, the manufacturer can assign item reference numbers to classes of objects that belong to it. Within the same class, similar objects can be distinguished by their serial number, which is a fundamental extension compared to the conventional barcode. In our dissertation, we will use the SGTIN-96 coding schema for our examples.

Data exchange between end users in the EPCglobal architecture is facilitated by EPCIS, which encompasses both data exchange interfaces and the data itself. The goal of EPCIS is to
enable disparate applications to leverage EPC event data via information sharing within and across organizations. Ultimately, this sharing aims at enabling participants of the EPCglobal network to gain a shared view of the disposition of EPC bearing objects within a relevant business context (EPCglobal, 2007b). Specifically, it also provides a standard query interface for retrieval of detailed EPC event data that is stored within the EPCIS repository. Much of the data provided by EPCIS consists of events such as observations of the object in particular locations within the premises of a company, as well as actions performed on it, e.g., packing, unpacking, shipping, and receiving. It may also include other attributes and sensor measurements associated with the object or its environment, e.g., price, quality, temperature, and humidity. The event data at the EPCIS level may be much richer than data provided at the ALE level because the EPCIS data model can answer more than the basic questions. Data stored in the EPCIS allows a number of additional descriptive fields to be specified per event and to provide annotations about the additional business context corresponding to each event. In our dissertation, we consider EPCIS as a repository that stores detailed information about an EPC event.

Finding information about a tagged object in an open system is a challenge because the EPC event data could be maintained by any EPCIS repository. To solve this problem, EPCglobal declares a component called ONS, which leverages the existing Internet standards and infrastructure. ONS uses the existing Domain Name System (DNS) for looking up/resolving information about an EPC (EPCglobal, 2008). The only purpose of ONS is to offer a lookup service, which provides no information except the address of the manufacturer’s EPCIS repository. It encodes complicated services in a standard way, which allows the service endpoint to be expressed as a URI (uniform resource identifier). The raw EPC sequence of bits will be resolved and converted, and the query and response formats must adhere to the DNS standards to enable the DNS to find the corresponding EPC data. After resolution of an inquired EPC, the root
ONS retrieves the local ONS and returns the address of relevant EPCIS servers. Figure 1.4 illustrates the ONS looking up process, which contains the following steps (EPCglobal, 2008):

1. A sequence of bits denoting an EPC is read from a 64-bit RFID tag.
2. The tag reader sends that sequence of bits to a local server.
3. The local server converts the bit sequence into the pure identity URI form as defined in EPC TDS (EPCglobal, 2010b).
   
   Example: urn:epc:id:sgtin:0614141.000024.400
4. The local server presents the URI to the local ONS resolver.
5. The resolver converts the URI form into a domain name and issues a DNS query. Example: 000024.0614141.sgtin.id.onsepc.com
6. The DNS infrastructure returns a series of answers that contain URLs that point to one or more services, e.g., an EPCIS Server.
7. The local resolver extracts the URL from the DNS record and presents it back to the local server. Example: http://epc-is.example.com/epc-wsdl.xml
8. The local server contacts the correct EPCIS server found in the URL for the EPC in question.

In some cases when the EPCIS address associated with the EPC, which is not known in advance due to the fact that it is the first time it appears in the network, or the EPC record is out of date though it is already exists in the network, the local ONS will consult the root ONS (in step 6). The root ONS contains the addresses of the local ONS, which is managed by an EPCglobal-associated organization. When the root ONS receives the query, it will identify the local ONS by recognizing the company prefix part of the DNS encoded EPC URI. Then it delegates the query to the identified local ONS server.

1.2.3 Motivation

Although the ONS can offer simply a lookup service, there is still a need for EPC Discovery Services (EPCDS) due to following reasons. In the EPCglobal specification documents (EPCglobal, 2010a), it describes the Discovery Services as a mean to locate EPCIS resources in the most general situations arising from multiparty supply chains or product lifecycles in which several different organizations may have relevant data about one EPC, but the identities of those organizations are not known in advance. Besides the functional and security concerns of the ONS that we mentioned above, it is also restricted by some other factors that make the discovery services necessary. We summarize the main differences between the EPCONS and EPCDS in Table 1.1.

The ONS is a lookup service useful for finding the address of the original EPCIS resource of a certain EPC tag. It does not address the issues of discovering the set of EPCIS data sources that may contain information about a particular EPC or a set of EPCs. Therefore, ONS is
the authoritative directory of information sources, which like a static pointer to point out an EPC’s head position in the supply chain, i.e., the manufacturer of the product. In the meantime, the EPCDS records all the history locations that an EPC has passed through, and thus it allows users to find the detailed, real-time, as well as the historical information about a certain EPC. The ONS usually stores the item’s class level information—for instance, the item’s ratings, reviews, price, availability, and specifications. However, the discovery service is able to store the serial level links for a specific item, such as traceability, authenticity, warranty registration, ownership, customization, and so on.

Table 1.1 Differences between EPCONS and DS

<table>
<thead>
<tr>
<th></th>
<th>Object Naming Services (ONS)</th>
<th>Discovery Services (DS)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Functionality</strong></td>
<td>• Point to current address of the manufacturer’s EPCIS, not designed for historical traceability information.</td>
<td>• Store historical addresses for every host, not only current state.</td>
</tr>
<tr>
<td></td>
<td>• Intended for relatively static records.</td>
<td>• Offer dynamic and up to date information.</td>
</tr>
<tr>
<td></td>
<td>• Stores class-level, not serial level, record.</td>
<td>• Handle serial level information for individual item.</td>
</tr>
<tr>
<td><strong>Scalability</strong></td>
<td>• Centralized service, limited by root ONS.</td>
<td>• Global scale distributed architecture.</td>
</tr>
<tr>
<td><strong>Robustness</strong></td>
<td>• Less robust, centralized service, limited by root ONS.</td>
<td>• More robust, distributed architecture.</td>
</tr>
<tr>
<td><strong>Services</strong></td>
<td>• Only specifies how to query ONS, not how to publish; it is merely a lookup service.</td>
<td>• Support different query types, one-off and standing.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Support publishing, multi-parties for one item.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Support complex service, active pull and push data.</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>• No authentication or authorization mechanisms.</td>
<td>• Offer mutual access control enforcement for each party.</td>
</tr>
</tbody>
</table>

In addition, the ONS is a centralized service that was designed based on the DNS infrastructure. Thus, this infrastructure is not a scalable solution as the number of EPC growing
rapidly in the IoT and EPCglobal network. Additionally, since the root ONS is not replaceable, it is not robust enough once the errors and attacks occur. Moreover, from the functionality and service perspective, the ONS specification only indicates how to query, but not how to publish, while the discover services should support both basic services and advanced services. Some studies also indicate that the ONS, from the security perspective, is not a good skeleton for building the discovery service (BRIDGE, 2007a), because the ONS does not support authentication or authorization for queries, while the EPCDS is located at the perfect place to enforce the security mechanism. In conclusion, the EPCONS and EPCDS should co-exist and serve different roles in the EPCglobal architecture.

Furthermore, EPCglobal claims that the root ONS runs by itself, whereas it is actually managed by a particular company based in a certain country. In this scenario, the functionality and reliability of the root ONS can be easily compromised by hostile companies or countries. Thus, for users in other countries, the ONS will not be a trusted system. Some studies (Evdokimov et al., 2008) have also indicated the possible attack model towards the traditional ONS architecture, e.g., unilateral ONS blocking and traffic eavesdropping and analysis.

As we can see from the discussion, the ONS is a lookup service for locating the address of the manufacturer of a specific item, which is part of EPCDS’s functions, but has some inherent deficiencies. Therefore, the secured Discovery Service is needed for providing more flexible and comprehensive services.

A number of EPCDS models have been conceived, but most of them are in the very early stage and can hardly fulfill the EPCDS design requirement, especially in the global scaled, highly dynamic real world environment, which involves a large amount of tags, and organizational entities, and machines. The number of tags is in the billions, and thus the data volume associated with the EPC event is extremely large. It is necessary to analyze current preliminary EPCDS
models in our dissertation and explore their strengths and weaknesses. Few works have been
done for this purpose, since EPCDS design is a relatively new topic. None of the current model
can fully meet all design requirements. The designs can be grouped into three categories. The first
group of designs (EPCglobal, 2008; BRIDGE, 2007a; Afilias, 2008; Polytarchos et al., 2008;
Khair et al., 2014a; Khair et al., 2014b; Muller et al., 2010) focused more on basic lookup and
publishing services, but overlooked advanced services such as standing query and message pushing
services. They address neither network performance nor security issues. A second group of designs
(Fabian, 2008; Xu et al., 2011; Dahbi et al., 2013; Paganelli and Parlanti, 2012; Liu et al., 2014;
Shrestha et al., 2010) adopted P2P technology to address core network design requirements but did
not drive to specific performance requirements such as query accuracy, network independency and
extensibility. The third group of designs (Grummt and Müller, 2008; Cantero et al., 2010;
Jakkhupan et al., 2010; Fabian et al., 2012; Kerschbaum and Chaves, 2012; Shi et al., 2012a; Shi
et al., 2012b; Dahbi et al., 2013; Kywe et al., 2013; Shi et al., 2013; Dahbi et al., 2014; Torvekar
and Li, 2014) addressed traditional security issues such as access control, but did not resolve specific
EPCDS security issues such as the mutual anonymous authentication, key publishing control, or
interdependent security issue.

A comprehensive EPCDS design requirement should be systematically summarized from
the extensive literature, and explicitly described. Then, a more sophisticate EPCDS network should
be built based upon a peer-to-peer enabled framework, to enable multi-functionality, and support
better network performance. Moreover, an integrated security countermeasure is needed to offer
mutual anonymous authentication, key publishing security control, and a federated access control
mechanism. Our ultimate goal is to satisfy all these design requirements.
1.3. Research Framework and Contributions

1.3.1. Research Framework

Motivated by the above observations and discussions, this dissertation proposes a framework that allows a versatile EPCDS model design and also enhances its networking capability and ensures its security. The framework is shown in Figure 1.5, where the research issues are organized by the design requirements of the EPC Discovery Service. These are grouped into three categories: EPCDS functionality and service, EPCDS network performance and EPCDS security and privacy.

Category I: EPCDS functionalities and services

- Problem 1: How to provide basic EPCDS functionality and service efficiently and effectively?
- Problem 2: How to provide advanced EPCDS functionality and service?

Category II: EPCDS network performance

- Problem 3: How to ensure the accuracy of EPCDS results?
- Problem 4: How do the EPCDS network performances meet the real world usage environment?

Category III: EPCDS security and privacy

- Problem 5: How to provide the mutual anonymous authentication for all the network participants?
- Problem 6: How to resolve the interdependent security issues?
- Problem 7: How to secure the key publishing procedures?
A number of studies have contributed to the research of EPCDS design, and most of them focus on solving the EPCDS functionality and service issues, especially to provide basic services (problem 1). However, they rarely address the issues relate to the advanced services. Thus, a better solution that can support advanced services (problem 2) is needed. Another group of designs tries to solve part of the network performances, whereas it can hardly fulfill all the requirements such as the query accuracy (problem 3). Most of them focus on improving the network scalability, while they overlook other requirements. Hence, we need a comprehensive EPCDS design to enhance the comprehensive networking performances (problem 4), such as the query accuracy, scalability, availability and reliability, independency and extensibility. Moreover, in the EPCglobal network context, each entity, whether human or machine, needs to be authenticated. Before the entities establish any business partnership and trust, the mutual anonymous authentication is a prerequisite (problem 5). Thereafter, entities are able to exchange their information based on their authentication results. An authorization is enabled by the relationship-based federated access control model (problem 6). In the meantime, the anonymous
authentication brings out the key publishing security issue, which needs to be resolved by the key publishing control mechanism (problem 7).

1.3.2. Contributions of this work

To address the above issues, this dissertation focuses on designing a new EPCDS model that can support all the required features. Figure 1.6 labels the problems and points out the corresponding solutions and chapters in our work. In the following sections, we will detail the two groups of problems and provide an overview of the solutions in this work.

The contributions of this dissertation are many-fold:

- First, this dissertation provides extensive discussion and analysis of selected typical EPCDS design models that were introduced by previous studies.
- Second, the detailed design requirements for the EPC Discovery Service system are discussed in a systematic fashion.
- The existing EPCDS model is discussed and summarized in relation to the EPCDS design requirements.
- The presentation of a P2P-based EPCDS design, which takes the comprehensive design requirements into account.
- A prototype of a new proposed EPCDS model has been implemented and evaluated. The prototype presents empirical evidence for the feasibility of core ideas of the new design.
- A security and privacy analysis of RFID and EPCglobal in general is presented for the EPCDS security and privacy model design preparation.
- It is the first study that addresses the resulting key publishing security issue and the interdependent security issue.
- A novel relationship-based federated access control model, which includes the federated mutual anonymous authentication and fine-grained authorization mechanisms has been introduced for the security and privacy purpose.
- An EPCDS source data ontology has been developed, which will facilitate the future works in this specific research area.

![Figure 1.6 Research Problems and Solutions](image)

1.4. Organization of the Dissertation

The rest of the dissertation is organized as follows. In Chapter 2, we review the main research that is related to our work, including background and preliminary technologies that we will adopt in the system design, particularly focusing on the selected typical EPCDS models. In Chapter 3, we collect the requirements that EPCDS should fulfill, and the current EPCDS models
will be analyzed and compared by using those requirements as a foundation. In Chapter 4, we introduce a new proposed P2P-based EPCDS network model. In Chapter 5, an EPCglobal network security and privacy analysis is presented, and a novel relationship-based federated access control model will be described. Finally, in Chapter 6, we conclude our work with a summary and will discuss remaining issues and future research directions.
Chapter 2: Background

2.1. RFID and EPCglobal Network

As described, RFID is an emerging wireless technology that uses tiny IC chips to uniquely identify, capture, and transmit information from tagged objects to enterprise systems via radio waves. A simple RFID sub-system includes two components: hardware (tags, readers and antennas) and the software systems (user-interface, middleware, and database system) (Finkenzeller et al., 2002). In general, there are three key elements: the RFID tag, the RFID reader, and the back-end system. An EPCglobal network is comprised of many RFID sub-systems from different stages of supply chains. The EPCglobal network consists of RFID sub-systems, EPC middleware (ALE), Object Naming Service (ONS), the EPC Information Services (EPCIS), and the EPC Discovery Service (DS). Each component plays a unique and important role in the EPCglobal network, which was introduced in Chapter 1.

2.2. EPCDS Models

Although the EPCglobal discovery service has not been standardized, several studies have been devoted to the development of this area. One group of designs was based on the non-P2P approach, while another group of models was a P2P-based design. However, most of them were aiming to build a lookup service by establishing contact between client and information resources, and thus they are limited to providing more advanced services. In addition, those non-P2P based approaches paid little attention to the networking designs, and thus have low scalability. Also, for most models, the final result accuracy cannot be ensured. Factors like availability and reliability, independency and extensibility are rarely discussed. Regarding
another aspect, although some works start to explore the EPCDS security and privacy issues, it is still a common gap for EPCDS design. Currently, few models consider integrating the authentication and authorization mechanisms, and the client and resource confidentiality can hardly be guaranteed either; in particular, the mutual anonymous authentication mechanism is missing, the key publishing is out of control, and the interdependent security has not been addressed yet.

A detailed analysis and assessment for selected EPCDS models will be presented in the next chapter.

2.3. Peer-to-Peer Computing

Peer-to-Peer (P2P) computing enables sets of dynamic network participants to share resources and to exchange information with each other directly via the overlay network.

In general, P2P systems can be classified as centralized and decentralized systems. A centralized P2P system usually has a central directory that indexes all the shared data. When it receives queries from users it will then learn the target resources from the index. On the other hand, the decentralized P2P system does not maintain a centralized directory, and instead, it distributes all the indexes into each peer node. The decentralized P2P systems can be further classified into structured and unstructured ones. The structured P2P systems allocate data into specific nodes that are calculated by certain algorithms. It has several key characteristics such as self-organization, robustness, and load balancing (Ding et al., 2005) that demonstrate better scalability and reliability. Normally, it offers no limitation on the number of nodes or the size of the network, and prevents the single point of failure problem. CHORD (Stoica et al., 2001) and CAN (Ratnasamy et al., 2001) are the classical structured P2P systems. Generally, they are able
to answer a query within a small number of hops. The unstructured P2P systems place data into nodes randomly, and thus they are not aware of the data’s location. Gnutella (Adar and Huberman, 2000) is the typical unstructured P2P system, which used the flooding search method to locate the desired resources. Additionally, according to the degree of decentralization, a P2P system can be classified as a pure or hybrid type. In the pure decentralized P2P network, each node plays an equal role, and none of them are more important than the others. In a hybrid system, however, some nodes serve as super peers that index data from their adjacent nodes.

In our proposed design, the structured decentralized network structure will be adopted, due to the fact that the nodes in EPCglobal network are not from a tightly transient population, which needs a scalable solution and efficient routing method to maintain good network performance.

### 2.4. Access Control Methods

Access control uses authentication and authorization modules to restrict the actions that a subject can take on an object. The common access control model includes Discretionary Access Control (DAC), Mandatory Access Control (MAC), and a Role-Based Access Control Model (RBAC). The DAC model has been used in many places, such as Windows OS. It issues access permissions based on a subject’s identity and needs to know the user, process, and groups to which they belong (Yuan and Tong, 2005). Bell and LaPadula (1975) proposed the MAC, which protects objects according to the labels that are associated with the subject and object. It is usually used by government or military departments, because the security is enforced by the system, and none of the subjects or objects can make policy modifications or access control decisions. RBAC is a popular approach that has been widely discussed (Sandhu et al., 1996; Bhatti et al., 2004; Xu
et al., 2004). Essentially, it is a special form of the DAC mechanism, which manages the access control based on different business roles of the subjects. The role is static and expensive to alter once it is assigned.

Attribute-based Access Control (ABAC) is a new paradigm that has been studied in recent years, e.g., Damiani et al. (2005), Shen and Hong (2006), Coetzee and Eloff (2007) and Hur and Noh (2011). It is composed of a policy set and architecture model that enforces the policies. Three attributes are defined: the subject attributes, resource attributes, and environment attributes. An attribute is anything that defines the identity and characteristics of a specific entity, such as a subject’s name or job title, object’s title or ownership, and current date and time of the environment. However, it lacks semantic interoperability, which restricts the application of ABAC in an open distributed system, and thus there is a need for the ontology and semantic mapping to support the accurate access control decision (Shen, 2009). The ontology and semantic based access control model has been discussed in some works, such as Shields et al. (2006), Pan et al. (2006), and Elahi et al. (2008). Although EPCDS involves a huge amount of complex EPC event data that needs to be taken care of by an ontology-based semantic authorization model, none of these studies has been done in this application domain.

The relationship-based access control and federated access control are considered to be a better fit for the cross-domain information sharing. The relationship-based access control has been studied in some previous works, e.g., Giunchiglia et al. (2008), Fong (2011) and Bruns et al. (2012). Federation represents the situation where multiple organizations are involved and collaborate together to act like one unit for outsiders, though each organization is an independent domain in terms of security, and thus has limited trust in others. Some works have been done in the federation authentication area, e.g. Security Assertion Markup Language (SAML, 2005) and
OpenID (2007), and the federated authorization area, e.g., Ardagna et al. (2010), Alam et al. (2011), Decat et al. (2013), and Demchenko et al. (2014).

With respect to these works, this dissertation further studies the interdependent security threat in the context of the EPCDS network, and proposes a relationship-based federated access control model that contains both relationship-based and federated authentication and authorization solutions.
Chapter 3: Review of Current EPCDS Models

EPCDS development is a cutting-edge research area. Although the EPCglobal discovery service has not been standardized, several studies have been devoted to the development of this area, for example, EPCglobal architecture (EPCglobal, 2010a), ID@URI approach (Främling, 2002; Huvio, et al., 2002), the RFID Directory Service and RFID Query Relay models (BRIDGE, 2007a), Service Location Protocol (SLP) model (Afilias, 2008), Service Lookup Service (SLS) model (Polytarchos, et al., 2008), and the Aggregating Discovery Service (ADS) model (Muller et al., 2010). One group of designs was based on the peer-to-peer approach, such as DHT-P2P model (Fabian, 2008), P2PONS (Xu et al., 2011), P2P RFID resolution framework (Shrestha et al., 2010), and Distributed P2P DS (Lorenz et al., 2011). Two studies, Polytarchos, et al. (2010) and Evdokimov et al. (2010), have attempted to assess the appropriateness of selected models, but they did not cover the newest models.

Polytarchos et al. (2010) examined and tested the Service Lookup Service (SLS) model under the simulation environment. They focused on providing a simulation method by using OMNet++ for evaluation of discovery services in terms of efficiency, scalability, and performance. Their results show that the SLS is able to increase the efficiency and scalability to some extent. In addition, their work illustrated a good approach to simulate an environment to test the performance of different EPCDS models. Although they briefly discussed several different discovery service models, there is no detailed analysis or comparisons. Moreover, it did not assess each model, nor provide hints for future designs.

Evdokimov et al. (2010) is a more comprehensive review work. They listed several existing models, and selected five of them for a detailed description. In our dissertation, we discuss eight important models in detail, five of them are non-P2P approach based models and the other three are P2P based models. Some newly proposed models have been included and
evaluated, e.g. Aggregating Discovery Service (Muller et al., 2010), P2PONS (Xu et al., 2011), P2P RFID resolution framework (Shrestha et al., 2010), and Distributed P2P DS (Lorenz et al., 2011). Some atypical models have been excluded, for instance, the EPCglobal specification did not provide a more concrete design and specification, while ID@URI cannot integrate with EPCglobal architecture, and thus has the standardization issue. The evaluation framework of Evdokimov et al. (2010) was derived from the ISO/IEC 9126 standard that is an international standard for the evaluation of software quality. However, the criteria of this software quality standard are not enough for the EPCDS assessment due to its specialty. Hence, we modified and extended this assessment framework, and propose a new classification of the design requirement that is service centric and specifically aims to facilitate the future design of EPCDS.

We summarize the design requirements for EPCDS from four earlier studies (BRIDGE, 2007b; Evdokimov et al., 2010; Lorenz et al., 2011; Kywe et al., 2012) and expand and categorize them into three sets: functionality and service, network performance, and security and privacy requirements. We classify the services into two different categories, and use some scenarios for illustration. Compared with their eleven technical functional requirements, our proposal considers both the network design and functionality design issues. According to our review, this chapter provides a better understanding and indication for the follow-up studies. Thereafter, a possible solution that is based on P2P infrastructure and relationship-based federated access control model has been introduced in the following chapters.

3.1. EPCDS Design Requirements

As the Discovery Service is required but underdeveloped, a discussion about the design requirements is necessary. Although there are limited works on this topic, the following studies
still provided us some clues. For example, the EPCglobal document (EPCglobal, 2010a) has indicated that the discovery services should include the following high level responsibilities:

- Facilitate visibility by providing a lookup mechanism to help find multiple sources of information related to serial level unique identifiers, particularly when that information is provided by multiple parties, is commercially sensitive, and/or not published in the public domain;
- Provide a set of one or more URLs, each accompanied by an indication of the type of service to which they correspond; such service types may indicate EPCIS interfaces, web pages, web services, additional discovery services as well as other kinds of services;
- Provide a means to allow parties to mutually identify and authenticate each other;
- Provide means to share information necessary for authorizing access to EPCIS service listings and EPCIS data, and provide means to securely pass authorization rules among parties;
- Provide a cache for selected EPCIS data for the purposes of resilient traceability or avoiding unnecessary cascading of queries.

These responsibilities present the basic high-level framework of what tasks the EPCDS network should fulfill. BRIDGE (2007b) was based on an online survey that showed various expectations of what a discovery service should do. This work was mainly from the client’s perspective and focused on the lookup service. Some studies, such as (Rezafard, 2008; Ilic et al., 2009), concentrated on discussing one requirement only. Muller et al. (2010) categorized the requirements for discovery service according to the ISO/IEC 9126, but simply listed the requirements, lacking a basic description and explanation. Other studies (Fabian, 2008; Kürschner et al., 2008; Främling et al., 2006) have drawn some specific lines and identified several issues
that are highly relevant to the EPCDS design. However, in our opinion, these listed requirements are not systematic and completed, and thus we identify some new requirements that are important but not mentioned in previous studies. We summarize and classify those requirements along with the ones that are provided by the earlier literatures from the perspective of EPCDS design. We believe there are three important factors that need to be met: the functionality and service design requirements; the network performance design requirements; and the security and privacy requirements.

Some of the functionality and service requirements have been mentioned in the previous studies, but not in a systematic manner. We further classify those requirements into two categories: basic services (BS) and advanced services (AS). The BS provides the most basic and fundamental services. From the clients’ perspective, it means the one off querying, which will find out the existing historical data and the current status of the corresponding item, and provide that information to the clients. For information resources, it contains the following functions: publishing and multi-party publishing services. These actions should be regulated by the security policies and enforcements, and match their privileges. The AS, compared to BS, is able to provide more complex and advanced services, such as standing query and pushing service. The standing query will notify clients the future updates. Clients need to do nothing but subscribe the desired information about certain items, and thereafter they will receive real-time updates automatically and dynamically. The pushing service enables the information resources to push the data to the users, even without their subscription. It does not require any initiation from users, and the data can be delivered to the corresponding parties.

For better explaining the situation, we identify four typical scenarios to exemplify different EPCDS functionality and services:
• **Scenario 1 (S1): Information Query.** When clients want to know the relevant information about certain items, they need to perform one off query that outputs static information of products’ current status. Clients submit a query to the EPCDS, the EPCDS will find out the EPCISs that have the corresponding information about this item, and then return those data to clients; for example, consumer C wants to purchase product P. C will query P’s EPC from the discovery services, the EPCDS will search this item and return the corresponding data or the addresses of EPCIS to C. As a result, C will receive the information about P’s location, price, status, temperature, produce date, expiration date, etc. The BS can serve this scenario.

• **Scenario 2 (S2): Information Publishing.** In this case, information resources want to publish the related information about the items into the network. Assume A, B, C are three organizations that are located in the same supply network, and they all maintain records about product P. Thus, every organization can publish information about P to the EPCDS.

• **Scenario 3 (S3): Inventory Subscription.** In this case, the clients want to know not only the static information of a product’s current situation, as described in scenario 1, but also the dynamic information continuously, such as future updates of a certain product. Hence, the system should allow the clients to subscribe their needed data. For instance, the downstream company D subscribes the inventory information of its upstream company U. The EPCDS is able to gather the related inventory information from the U’s EPCIS, and provide the data to D when U’s inventory level reaches a certain threshold. The AS will support this scenario.

• **Scenario 4 (S4): Product Recall.** In this case, the network should allow the organization to push the product recall notification to the network. The EPCDS will
then broadcast this message, and find the last node that the EPC has passed, and then push the recall message to the corresponding party. However, if the record has shown that this EPC is in the recycle phase, the recall message will be discontinued. On the other hand, when the consumers stop using the products, or for any other reason, they can also terminate any future AS updates for this product. The predictable result is that the query from users will be reduced dramatically, and users can obtain up to date information about their product. The AS is able to fulfill this scenario.

To meet different users’ expectations and operational requirements, a discovery services model should be carefully designed and implemented considering and meeting many requirements. We briefly describe and comment on the requirement below.

Firstly, the functionality and types of services a system can offer is an important dimension for EPCDS. We identify five design requirements in this group.

- **Requirement 1 (R1): Basic Lookup Service.** When clients query a unique identifier (EPC key) and its EPC event data via the DS query interface, the DS should be able to find out all the addresses of the information resources (EPCIS repositories) that related to this key.

- **Requirement 2 (R2): Publishing Service.** DS should not only support read functions (e.g., querying), but also write functions (e.g., publishing). Therefore, the information resource should be able to create or add new EPC records to DS network. Each record is published by only one entity.

- **Requirement 3 (R3): Multi-Party Publishing.** The EPC event data associated with a specific EPC key may be maintained by more than one organization, and thus all these parties should be allowed to publish this EPC key into DS, i.e., multi-party publishing. Clients use the published EPC key to locate these EPCIS and data.
However, people have realized that the BS cannot satisfy the needs, because it is essentially similar to the ONS, though it is more powerful. Thereafter, for the sake of users’ benefits, particularly for the organizational users, and improving the competitiveness of the whole supply chain, the discovery services should provide more advanced services (AS). The AS include the standing queries that need publishing and subscription mechanisms to notify clients of future updates. The users only need to initiate the setup, and it will automatically pull in the subscribed information; for instance the scenario 3 represents this type of services. The AS offer more complex and dynamic services, which do not need users’ extra efforts. For example, information resources can push important and emergent information to notify clients automatically and dynamically, even without their subscriptions; e.g., in scenario 4, the users are able to receive the up to date recall notifications. Thus, we need requirement 4 and 5.

- Requirement 4 (R4): Subscription Service. The DS network needs to support the standing query, which means when a user subscribes certain EPC event data, the DS should be able to notify the subscriber with all the future updates.

- Requirement 5 (R5): Pushing Service. The DS should enable the information resources to push the data to all the corresponding parties, when there is important notification that needs to be sent out, even without a subscription.

In addition, for the network performance design aspect, several requirements in terms of the network’s capability and performance need to be considered, inquiry accuracy, availability and reliability, independency, extensibility and scalability.

- Requirement 6 (R6): Inquiry Accuracy. When a client queries the EPCDS, the network must be able to guarantee and prove that the final result is complete and accurate.
• Requirement 7 (R7): Availability and Reliability. This requirement seeks for the robustness, fault tolerance, and failure recovery features. EPCDS is under a highly dynamic and diverse environment, which requires robustness to resist the single failure problem, random errors, and attacks. In case the failure occurs at one or more nodes, the whole network should still be able to functional properly. In addition, certain self-recovery capabilities are desirable. For example, if one node fails to deliver the address of the target EPCIS, this task should be able to compensate by other nodes, e.g., the neighbor nodes. The bottom line is that the failure of nodes does not affect the functionality of other nodes and the entire network.

• Requirement 8 (R8): Independence. Because the EPCglobal network involves various participants in different locations that may use different information systems, EPCDS needs to be independent from other entities by offering a generic interface to interact with them. If there is any change to these components, it should not affect the functions of the EPCDS.

• Requirement 9 (R9): Extensibility. EPCDS needs to serve as a flexible underlying infrastructure that can integrate with other components, e.g., access control and quality of service (QoS), etc.

• Requirement 10 (R10): Scalability. Scalability is the most important requirement for EPCDS design, because the system is expected to be deployed globally, it should be able to handle large amounts of requests and data, and deal with increasing numbers of users, objects, and machines. For this purpose, EPCDS needs to demonstrate excellent efficiency and load balance characteristics.

Finally, it is obvious that the EPCglobal system needs some security mechanism to regulate the behavior of the network participants. EPCDS is a suitable component to integrate the
security enforcement. However it raises some key requirements that are specifically for the EPCDS security design.

Firstly, the system users vary from individual consumers to organization participants, who do or do not have a business relationship with the target information resources. Under this dynamic environment, different subjects will obtain different privileges while accessing a certain target object. An object refers to the desired information that is requested by a subject. This could be any element of the database, e.g., a table or an attribute of that table. The subject refers to the entity that intends to access the object. Apparently, figuring out the business relationship between the subject and object is crucial for the access control decision making. For example, if the client is an individual consumer, it may be granted the lowest privilege that can only access the public information of the product. However, if the client is a business partner, e.g., the manager of its downstream company, more privileges will be granted.

Secondly, it is also important to offer an option that allows the client and the information resource exchange data while keeping their identity undiscovered. When two entities do not have any partnership, the trust has not been established yet, this anonymous mechanism is considered as a desirable feature. And of course, they will always have the option to turn anonymity off by disclosing their identities.

Thirdly, because EPCDS is an open system that has no control over the key publishing procedure, it creates a vulnerability that every entity, no matter legitimate or malicious, can publish data into the network. If a malicious entity publishes the forged EPC key into the DS Item Table, it should be able to disguise its identity and pretend to be a real supply chain partner. This is because under the anonymous authentication scenario, the identities of service requesters and service providers will not be disclosed to each other. The supply network partners rely on the intermediate broker, i.e., EPCDS, for authentication. EPCDS uses the DS Item Table to identify
the partnership between these entities. This procedure will be explained further in the following sections. A false business partner is a serious threat for the EPCDS network and its participants, because it usually can obtain higher privileges when querying sensitive data. In order to prevent this threat, the EPCDS needs to provide a key publishing control mechanism to be sure of the authenticity of the publisher.

Fourth, establishing a uniform security policy for all the network participants is not applicable under the EPCglobal context as each organization controls its own EPCIS database by defining its own security policy. For example, organization “A” may allow an individual consumer to access some of its information, while organization “B” may deny all the query attempts from individual consumers.

Fifth, another aspect the security enforcement should also be deployed at the network level, though each organization has its own local EPCIS security rule. The network level access control model needs to read and execute the local security policies in order to make the authorization decisions for different objects.

Next, since the EPC event data is the major object of protection, and it is the data-centric and complex object, a fine-grained security model is a must to handle such complicated objects in a distributed environment. It must ensure that each different portion of the EPC event data meets the various security rules. This is due to the restriction that these businesses’ sensitive data should be accessed only by the client who has a legitimate need-to-know and is permitted by the information owners.

Finally, the biggest risk of the EPCglobal network is that the organization is exposed to a wider supply network context, but the access control mechanisms applied within itself are just one part of the security it demands, which causes these participants to suffer from the higher level risk, called an interdependent security threat. The interdependent security threat is defined as the
risks faced by any one agent depending not only on its choices but also on those of all others (Kunreuther and Heal, 2003). Under the EPCglobal context, it means that the security of one single organization not only depends on its own security rules or access control mechanism, but also is highly related to the security decisions of its partners with whom they have business relationships.

The DS node needs to collect data from the data repositories that are managed by different organizations, which declare their own security rules independently. These security rules are designed based on their own security requirements and needs. This brings a significant security problem during the sub-results aggregation phase, because some of the sub-results may overlap with each other. Although the sub-results are extracted from different domains, they are still carrying relational data. Because all of these EPC event data refer to one specific EPC key, it means that they are the attributes for one certain object. Figure 3.1 illustrates an example of the interdependent security threat. Here r.a and r.b denote the sub-results from repositories IS.a and IS.b, respectively. If companies A and B have business relationships in the matter of a certain object EPC e1, the information of e1 that are stored in IS.a and IS.b are relational data. In this case, both IS.a and IS.b are carrying the same data a1 and a2. When a client looks up EPC e1, the EPCDS will collect all the potential sub-results, which are r.a and r.b, from IS.a and IS.b. However, the final result is the union set of the aggregation of all sub-results, i.e., final-result = \{r.a \cup r.b\}. This will, unfortunately, cause a severe security problem if their security rules have conflicts. For instance, IS.a allows all of their data being disclosed to its client, while IS.b considers that data as confidential data, and thus restricts all the clients to access it. Although the client cannot get a2 from IS.b, the final result will still contain a2, because it is contributed by IS.a. As a result, the security goal of IS.b is compromised. In this scenario, it requires an access control model that takes all the entities’ security demands into consideration.
In summary, EPCDS encounters a very complex and difficult situation, and the traditional security countermeasures are not suitable under this circumstance. The proposed security model must be able to meet the following security and privacy requirements efficiently, and it has to be a context-aware model.

- **Requirement 11 (R11):** EPCDS needs to be able to identify different business relationships. A client’s identity is usually unknown because EPCDS users are highly diverse and dynamic; thus, it is hard to identify the business relationship between the client and the information resource, whereas the business relationship is the critical evidence for the information resource to enforce the corresponding security countermeasures.

- **Requirement 12 (R12):** EPCDS needs to be able to offer an anonymous mechanism. Because the EPC event data is sensitive, some entities may wish not to reveal their identity before establishing partnership.

- **Requirement 13 (R13):** Key Publishing Security. The EPCDS needs to provide a key publishing control mechanism to be sure of the authenticity of the publisher.

- **Requirement 14 (R14):** EPCDS needs to support database level security. Each organization publishes information independently, and thus they will need to have the ability to manage the security for their own EPCIS, instead of setting up a holistic
network security plan. That is, the policy repository is managed by each business entity independently.

- Requirement 15 (R15): EPCDS needs to provide network level security. Although there is no way to define a holistic security rules for the entire network, the security enforcement still needs to be implemented at the network level. The network level security component needs to interact with the security policy sets defined by different organizations.

- Requirement 16 (R16): EPCDS needs to execute a fine-grained access control at the attribute level.

- Requirement 17 (R17): EPCDS needs to resolve the potential interdependent security threat. An interdependent security threat is the most significant issue, because the target data are pulled out from each independent data repository, while these data are related to each other.

3.2. Selected EPCDS Models

At present, there are several proposed discover services models can be found in open literature. We select eight typical designs for detailed discussion. Five of them are non-P2P approach based models, which include RFID Directory Service and Query Relay models that from the BRIDGE project, Service Location Protocol (SLP), Service Lookup Service (SLS) and Aggregate Discovery Service (ADS). The other three are P2P based models, which include DHT-P2P, P2PONS and P2P RFID resolution framework (P2P-RFID). After that, we analyze the advantages and disadvantages of all the current EPCDS designs based on the design requirements, and thus provide a better understanding and inspiration for future directions.
3.2.1 BRIDGE Project

BRIDGE (Building Radio Frequency Identification for the Global Environment) is a European RFID project running over three years and partly funded by the European Union. The objective of the BRIDGE project is to research, develop, and implement tools to enable the deployment of EPCglobal applications in Europe. According to BRIDGE, the primary role for discovery services is to provide a mechanism to allow computer systems and application software to find the network addresses of information resources that provide more detailed information about an individually identifiable physical object, especially where those resources are distributed across multiple organizations within the object’s supply chain or lifecycle history. Discovery services are intended to be lightweight and primarily provide links to data resources such as EPCIS instances where more detailed event information can be retrieved directly via secondary query.

Several interaction models have been proposed and discussed in their document (BRIDGE, 2007a). After the comparison, four candidate models are selected: the directory of resources model, the notification of resources model, the query of propagation model, and the Meta client model.

- The directory of resources model. In this model, the information provider, EPCIS, will publish the relevant information, e.g., EPC and reference number, into the intermediary—EPCDS. Clients will query and obtain the wanted information from EPCDS and then contact with the information resources directly. Some problems have been identified with applying this approach, for instance, most directory models assume that the information is either public or visible within a controlled user group. For the publication of EPCIS resources this will not be suitable, as data owners will desire more fine-grained controls over who can see the information on the directory, because of the business sensitivity of
that information. This model requires information resources to store their EPCs and the relevant references, e.g., URLs and services types, in the DS.

- **The notification of resources model.** In this model, clients register their target EPCs with the discovery services, EPCIS broadcasts its identity and the EPCs it holds, DS sends notification to the interested clients, and then the clients contact the information resources directly and obtain the information they need. This interaction model differs from the directory-based approach in one key manner. In the directory service, clients usually connect to the Discovery Service infrequently and query to DS over historical resource information. In this model, clients are assumed to be continuously connected to the network and respond in an event driven manner to new EPC resources. This model requires clients to store their requested EPCs in the DS.

- **The query propagation model.** This model needs the EPCIS to publish their information, EPCs, and the corresponding references, to the DS first. The clients will issue queries to the DS, and the DS will propagate the query to the resources that hold relevant information. The EPCIS will then answer the clients with detailed information that was requested. This model needs clients to fully trust the DS, and requires information resources to store their EPCs and the relevant references, e.g., URLs and services types, in the DS.

- **The Meta client model.** This model asks the clients register their query with DS, and the EPCIS will search for the proper query that it can serve and obtain the potential clients’ information from DS. The EPCIS will interact with the clients directly. Clients who use this model for querying may not be able to get their information immediately, because they have no control over the time and frequency that the relevant information resources spend for searching. Moreover, if more than one resource is involved, they may have
different searching schedules, and thus the clients cannot know when their query results will be completed. This model requires clients to store their queries in the DS.

The directory of resources and notification of resource models vary only in whether the client predominantly requests and replies or publishes and subscribes, and therefore, a combination of these two models to provide directory services with a notification capability for publishing and subscription operations will be appropriate. In addition, the query propagation model and Meta client model involve the clients sending requests via the discovery services to the relevant information resources, and they are integrated into one model—the query relay model. The two final proposed models will be discussed in more detail.

3.2.1.1. RFID Directory Service

Discovery services can be implemented as directory services that are concerned with highly fragmented sources of data and highly dynamic links between those fragments. The trace of previous hosts for a given object might be completely unique to that particular object and might never be repeated exactly for other objects of the same type nor fully known in advance when the object begins its journey. Discovery services can provide an element of robustness or redundancy to the process of gathering information, since their directories can return a list of links to multiple providers, whereas in their absence, the client would need to locate the original information resource, e.g., the manufacturer’s EPCIS, and then attempt to follow onward links from one organization to the next—if this information was indeed provided by the EPCIS. The major vulnerability of this link traversal approach is that if the EPCIS of any of the intermediate parties were unavailable, due to many reasons, e.g., temporary power or connectivity outage or permanent cessation of trading, it may become impossible to navigate the missing onward links to downstream parties. The Directory Service approach allows each organization to publish a link to a DS in order to avoid the risk of a single point of failure within the supply chain, provided
each DS itself has guaranteed availability; in this situation all other organizations remain reachable.

The proposed RFID Directory Service supports both the one-off query and standing query. Clients wish to receive all history record and future updates about the events received by an EPCIS repository, which match the clients’ specified query criteria. That requires a subscription request to be sent to the Directory Service from the client. Afterwards, whenever the EPCIS updates the information on the Directory Service, it will notify the appropriate clients regarding the change. The Directory Service contains a list of pairs of EPCs and EPCIS servers that can provide services and data. The process of discovery is as follows:

- The EPCIS informs the directory service of the EPCs it has information about;
- The client requests from the directory service the EPCIS that holds the data for an EPC;
- The client requests a service from that EPCIS.

For instance, in Figure 3.2, EPCIS 1, 2, and 3 register their information in DS. When the client looks for EPC 123, it will initiate a query to DS. The DS will search EPC 123 in its own record and respond to the client with the result. When client receives the answer from the DS, it will know who possesses the information about EPC 123, in this case, EPCIS 2 and EPCIS 3 are the target information resources. Thereafter, the client will contact the relevant EPCIS and get the result directly from them.
Figure 3.2 Directory Services Model

This model is considered as the most basic discovery service approach that fulfills the core functional requirements, e.g., the basic services (BS) R1, R2, and R3, as well as one of the advanced services (AS) R5. The pushing service R6 has not been mentioned. Although it did not touch the security and privacy issues in depth, it can be addressed by introducing some existing authorization and authentication mechanisms, e.g., public key infrastructures and digital signature. However, the confidentiality for neither the client nor of the information resource can be assured, because they will directly communicate and expose data to each other.

Other than these, the more significant problem for this model is that it does not address the issues that are related to the network performance requirements. The query relies heavily on the availability and reliability of EPCDS, while if the server is down, the client will not be able to perform any query and cannot obtain the EPCIS addresses. In this case, the single failure situation
will occur. In addition, the design did not discuss how to achieve the independence, scalability, and extensibility requirements.

3.2.1.2 RFID Query Relay

In the Query Relay Discovery Service scheme, the intermediary forwards the client’s queries to the appropriate EPCIS and the EPCIS in turn connects with the client in order to provide a service. The Query Relay architecture provides a more secure approach. The EPCIS may choose not to respond to a client’s query. As a result, the client is not capable of knowing whether there is an EPCIS available for a specific object. As for the standing queries, the subscription is propagated to the EPCIS, which in turn notifies the required clients for the appropriate changes. Given that information on the flow of products is commercially sensitive information, it is obvious that the companies will be reluctant to provide this information to anyone. Thus, in order for an EPCIS to appear on a result list of the EPCDS, the consent of the owner of the EPCIS will be necessary. In this model, the DS does not return any resource data to clients, and it propagates clients’ query to resources and the client must wait for them to respond. The same client query will be sent to all resources and the client has neither visibility nor control over which publishers will receive their query. In addition, the DS does not store fine-grained access control policies on behalf of resources. Authorization is done by each resource independently. Each resource can log all attempts to access their data, and is able to deny access to certain clients without making the client aware of the denial.
Figure 3.3 illustrates the example. First, the EPCIS registers their information to the DS. When the client needs information about EPC 123, it will generate a query to DS. DS will look up its record that relates to EPC 123, and find out which information resources maintain the record of this specific EPC—in this case, they are EPC 2 and EPC3. Then, the DS will communicate with the corresponding EPCIS directly without notifying the clients. After that, the DS will answer the clients with the information it collected. Compared with the previous model, the major difference is that clients will not have a chance to query the information resources directly, and the advantage is that the identity of the information resources will not be disclosed.

Compared to the first model, this approach implements an asynchronous request and response paradigm that requires the client to implement an extra interface to receive and aggregate the returning EPCIS responses. The same as the RFID Directory Service model, the Query Relay model is able to fulfill R1-R3.
This model can insure better confidentiality, because the client will not have the chance to directly query the information resource, and thus the information resource’s sensitive data, e.g., EPCIS address, will not be disclosed. On the other hand, since the client cannot gain any information about the EPCIS, the client’s data will be exposed to the other entities without the client’s awareness. The confidentiality of the client cannot be well protected in this model.

Also, this model did not address the important issues that are related to the network performance requirements, requirements R6 to R10, which is the same as the previous one. A client requesting information has no clue about the number of potential EPCIS that are in possession of information regarding the queried EPC, and thus does not how many responses it needs to expect. In addition, some EPCIS might have slow response times, deny a response to the query or be temporarily unavailable. Therefore, the client has to wait for an uncertain period of time until a timeout is reached. The accuracy of the result cannot be guaranteed, even if no more replies will be received. Similarly, the single failure issue may occur, and the model does not provide much discussion for the independency, scalability, and extensibility requirements.

3.2.2. Aggregate Discovery Service (ADS)

ADS (Muller at al., 2010) is similar as the BRIDGE’s Query Relay model. It forwards client queries to relevant EPCIS servers and returns the results to EPCDS, instead of returning the EPC event data directly to user. After getting replies from different EPCIS repositories, EPCDS aggregates their information and synchronously sends them back to the user. Figure 3.4 is a simple illustration of ADS.
First, EPCIS needs to register to EPCDS with the details concerning which EPC numbers they are handling, together with its URL address. Then, EPCDS stores the pairs of EPC numbers and EPCIS addresses in the lookup table. When a client sends a query to EPCDS with a specific EPC number, e.g., EPC 123, EPCDS will search its lookup table for the queried EPC numbers, finds corresponding EPCIS addresses and relays the client’s query to those EPCIS resources. Each EPCIS returns the query result to EPCDS according to the client’s privilege. Thereafter, EPCDS aggregates the all the sub-results that were returned from different EPCIS and displays the final view to the client, and so both client and EPCIS do not need to directly communicate with each other.

The difference between the ADS and Query Relay model is that the former needs to collect the sub-results from all the information resources and aggregate those sub-results to the final one, and present the final result to the client. The latter model will simply relay the client’s query to the information resources and client will receive the results from them.
This approach combines some of the advantages of the first two models. It supports the requirement R1-4, but not R5. A major improvement for this model is that the discovery services relays the client query to the respective EPCIS servers, which enables the EPCDS to ensure the inquire accuracy (R6), and also protects the privacy of the information resources. In the meantime, ADS can control the query process, enabling it to take remedial action upon non-responding resources, though like the Query Relay model, the client privacy is still an open question.

Additionally, the availability and reliability of this model is also unsure in this model, since the single point of failure may occur. Once EPCDS is unavailable, a client has no way to communicate with EPCIS resources as users do not have the chance to gain any information about EPCIS addresses. Scalability is another big problem of the ADS model. When the number of user queries increases, EPCDS needs to handle exponentially increased connections to EPCIS resources and processing of the results. Moreover, Muller at al. (2010) did not describe the authentication and authorization mechanisms in particular.

Overall, the ADS model provides the best interaction control between the client and information resources, though the networking performance gives rise to more concerns.

3.2.3. Service Location Protocol (SLP)

Service Location Protocol (SLP), also called Afilias, tries to solve following issues: unique identification of items in a world of diverse identifier authorities, backward compatibility with existing identification schemes, concerns regarding control of a single point of authority that is outside local boundaries, assurance of practicality, scalability and openness to competition in the provision of services, and trust and security of the system (Evdokimov et al., 2010). It employs an open web services protocol called the Extensible Supply-Chain Discovery Service
(ESDS), which is originally an Internet Engineering Task Force (IETF) standards track protocol that provides a framework to allow networking applications to discover the existence, location, and configuration of networked services in enterprise networks.

ESDS (Rezafard, 2008) refers to a lightweight referral service, whose sole purpose is to help a client to find one or more sources of detailed information, which is not an aggregator of detailed information. ESDS returns a list of URLs of information resources that contains detailed information about a particular object to an authenticated client. The client will only receive the URLs to information resources that have authorized the client to see their URL address for that specific object ID. The client can then contact each of those information resources in order to request more detailed information, which is outside the scope of ESDS.

Figure 3.5 Afilias Extensible Supply Chain Discovery Service

Figure 3.5 shows the ESDS work flow. Within the supply chain, each information resource (IR) will register to the ESDS first, when the client sends the query to ESDS looking for information about item 123, the ESDS server will provide the relevant URLs to the client, and let the client communicate with the IR. In some cases, when the client requests data outside the
supply chain, the ESDS server will send a request to the global ESDS server community that will be routed to other local ESDS servers. If any server has the requested data, the community will use a protocol, such as JXTA, for peer-to-peer communication between each ESDS server. The ultimate goal is to locate the target discovery server without dependence on hierarchical information and services, such as ONS or the underlying DNS. However, the details of implementation are not made public by Afilias (2008).

This model offers R1-R4, but not R5. Due to the similarity of SLP and the RFID Directory Service, they have the same weaknesses. In particular, the details of implementation of SLP are not available, so the achievements of networking performance, and security and privacy requirements cannot be determined.

3.2.4. Service Lookup Service (SLS)

SLS (Polytarchos et al., 2008) is a three-level hierarchical structure designed to resolve the deficiencies of the ONS and provide a more efficient and scalable counterpart. Similar to the BRIDGE’s RFID Directory Service model, which is less scalable, SLS is more scalable. The three layers are: Object Directory (OD), Object Services Directory (OSD) and Object-Service Information Server (OSIS). Figure 3.6 illustrates the lookup procedure. First, the clients send a request to OD that will respond to the clients by the address of the OSD. OD locates at the top level and stores information about the address of the specific objects. It provides the URL of the OSD by resolving the EPC, similar to the ONS. After obtaining the address of the OSD, the clients will contact it. The OSD replies to the clients by the address of the corresponding OSIS. OSD locates at the middle level and stores the information about the services associate with each specific object and the link to the OSIS. OSIS is located at the bottom level that holds the actual
object service information—optimally, every entity within the supply chain maintains its own OSIS. Clients will receive the final results by requesting them from the OSIS, when they get the address of OSIS from the OSD.

Figure 3.6 Service Lookup Service Query Procedure

This model is able to accomplish the basic services (BS) only, but not advanced services (AS). SLS also possesses the weaknesses of the RFID Directory Service, for example, no guarantee of client and resource confidentiality. However, one improvement of this model is the scalability due to an extra layer deployed to the network.

3.2.5. P2P Approach based Models

The Peer-to-Peer Approach based models include DHT-P2P (Fabian, 2008), P2P RFID Resolution Framework (Shrestha et al., 2010), P2PONS (Xu et al., 2011) and Distributed P2P DS (Lorenz et al., 2011). They are built on the same underlying P2P technology, and essentially similar to each other.

However, P2P RFID Resolution Framework, P2PONS and Distributed P2P DS are preliminary designs that still need some improvement. Shrestha et al. (2010) proposed the P2P based RFID Resolution Framework to deal with the challenge of resolving the data stored in the RFID tags to the actual location of the data in the network. There are several significant weaknesses in this approach. For instance, it is suitable for small and medium scale businesses,
but a large scale network. In addition, other than the lookup and publishing services, it has no
intention of discussing other advanced services. However mechanisms of this work for handling
nodes join, leave, and fail, and the authentication and authorization mechanisms are still
incomplete. P2PONS is a distributed object naming service based on P2P for EPC network that
are proposed by Xu et al. (2011). This P2PONS shows low lookup hops that are $O(\log(N))$ or
$O(1)$, low response time, and load balance. The major gap in this approach is the lack of security
considerations and the inability to offer more advanced services. Lorenz et al. (2011) introduced a
partitioning scheme and developed a prototype for distributing EPC among independent
discovery services. It provides companies with the ability to decide where to publish and store
their information. Therefore, the scalability and political issues are addressed. But more work is
still needed to consider how to provide more advanced services along with the security and
privacy issues.

Compared to these three P2P based approaches, DHT-P2P, which is designed by Fabian
(2008), is a more complete work. Hence, we will describe this model in detail and use it to
represent the P2P based group of models.

DHT-P2P is a structured P2P EPCDS model, which uses Distributed Hash Tables (DHT)
to offer high robustness to faults, avoid single points of failures as there are no root nodes, and
distribute responsibility and load among participants in a systematic way by means of a
prearranged topological overlay structure (Balakrishnan et al., 2003). One of the advantages is
that the DHT-based P2P approach can be integrated into an EPCglobal network application
landscape. In addition, DHT enables fundamental lookup functionality and can be suitable for
large scale and robust global discovery services.

Among those P2P based EPCDS designs, the Bamboo DHT-based Object-Information
Distribution Architecture (OIDA) (Fabian, 2008), shown in Figure 3.7, is the most typical and
complete one, which has been proposed and tested. Hence, we will select this DHT-P2P
describing its detailed working process, and discuss its strengths and weaknesses, while other P2P
based models are similar to it.

Figure 3.7 DHT-P2P Architecture

In the DHT-P2P model, each supply chain partner publishes its EPC associated event to a
local ONS—e.g., EPCIS 1 and EPCIS 2 publish their information to ONS 2 and EPCIS 3
publishes to ONS 1, etc.—and that is achieved by using a local peer client to insert a key pair
(key= \( h \) (EPC), value= EPCIS address document) into DHT, where \( h \) is a hash function such as
SHA-1. The cryptographic hash value \( h \) (EPC) can be used us a confidentiality enhancing
measure due to the one-way property, and also be used as a DHT lookup key. When an
information resource wants to publish information for a certain EPC, for example, the address of
a corresponding EPCIS, it will first create a document containing its name and the information,
and the cryptographic hash of the EPC; it also includes the version control information, time
stamps, and TTL. If a central CA is used for OIDA, a certificate signed by it could be added,
linking the information resource with its public key. In order to avoid single points of failure, a replica identifier, $r$, for the CHF input will be employed to achieve redundancy. The final document $d$ is then stored $r_{\text{max}}$ times in the DHT, $1 \leq r \leq r_{\text{max}}$. When clients ask for information about a specific EPC, they issue a request to one or many DHT gateways by using value $r$, until a copy of document $d$ is retrieved. DHT replies by sending $d$ to clients.

One supply chain may deploy more than one ONS that is according to the geographical base. The P2P based EPCDS system will use the DHT to look up information about the particular EPC for the client, and responds to the clients by the EPCIS addresses. Therefore, the clients are able to retrieve the relevant information from the corresponding EPCISs, e.g., in this case, clients retrieve information about EPC 123 from EPCIS 2 and EPCIS 3.

P2P based EPCDS usually focus on network performance design, rather than the functionality and service aspect. Most of them can only offer the basic passive services (BS), but not advanced service (AS). Since the client and the information resource will directly communicate and expose data to each other, their confidentiality can be assured. The most considerable improvement of this approach is that it provides a high level of scalability without relying on any centralized login servers or other third party service provider. Another advantage is that the P2P based methods do not contain any entity that within the distributed network has superior power or is more important than others. This enables the avoidance of the signal point of failure, and consequently improves the system’s fault tolerance, robustness, and availability and reliability, which are very important for the global based EPCglobal or IoT network. As a conclusion, we believe that a DHT-based P2P network technology is the most promising approach for the future EPCDS design, considering scalability as the predominant requirement. But we still need to eliminate its disadvantages, such as the unsure client and resource confidentialities, and the inability of offering the AS.
3.3. Summary of Current EPCDS Models against Design Requirements

Some works have been done to evaluate and compare different models, especially by Evdokimov et al. (2010) who developed a comprehensive framework based on ISO/IEC 9126 software quality evaluation standard to compare several discovery architectures. They provide some descriptions and assessment criteria from the functionality, reliability, efficiency, maintainability, and portability aspects. Focusing on guiding the future EPCDS design, we have employed and extended their framework and criteria into three categories: the functionality and services; network performance; security and privacy.

Since some of the existing works focus on functionality and network issues, while the other group concentrates on the security and privacy issues specifically, we will summarize and compare these two groups of works against the aforementioned design requirements separately. Our goal is to find out their strengths and weaknesses, and thereafter to benefit and guide the follow up design works.

3.3.1. Functionality and Network Performance

Over the past few years, several studies have been devoted to the development of EPCDS. EPCONS is an instantiation of EPCDS that is defined by the EPCglobal specification (EPCglobal, 2008), which is built upon a similar technology to the Domain Name Service (DNS). However, a security analysis by Fabian et al. (2005) indicated that DNS is not a good choice for EPCDS infrastructure. Moreover, the ONS is merely a simple lookup system that cannot fulfill the functionality and service requirements needed by DS.

The BRIDGE (BRIDGE, 2007a) is a comprehensive project, which proposed the RFID Directory Service and Query Relay models that offer the fundamental methods for key publishing
and query; however, it did not provide the detailed network design. Service Location Protocol (SLP) (Afilias, 2008) and Service Lookup Service (SLS) (Polytarchos et al., 2008) are two hierarchical discovery service models based on the Domain Name Service (DNS). They share the same limitation as DNS and ONS in that they cannot meet the multi-publishing and advanced service requirements. Border Gateway Protocol (BGP) based discovery service models (Khair et al., 2014a; Khair et al., 2014b) can show the most up-to-date EPC status, however it hardly provides advanced services. The Aggregating Discovery Service (ADS) model (Muller et al., 2010) introduced a promising result collection method, which we have adopted. However, the key publishing procedure and network design are not explained.

Some other designs were based on the Peer-to-Peer approach for improving the scalability. For example, Distributed Hash Table (DHT) based EPCONS models (Fabian, 2008; Xu et al., 2011; Dahbi et al., 2013) can support the scalable networks. However, they also inherit the restrictions from the ONS, e.g., no multi-publishing and advanced services, though Fabian (2008) is a more comprehensive study than (Xu et al., 2011; Dahbi et al., 2013) as it analyzed the critical network performance features more, such as availability and reliability. A DHT-based DS (Paganelli and Parlanti, 2012) focuses on service resolution, instead of the network design and security issues. P2P based hybrid architecture (Liu et al., 2014) briefly describes the query and key publishing services, whereas it lacks the depth for practical implementation. The P2P RFID Resolution Framework (Shrestha et al., 2010) demonstrated a great potential for scalability using a consistent hashing approach, but it lacks actual implementation. We extend this method by introducing detailed service procedures for EPCDS and further explain how to accomplish the proposed network and security expectations.

Table 3.1 summarizes and compares the past related studies according to the design requirements that we identified in Section II. Note that the table only considers the studies that
provided actual solutions, rather than those that discuss high-level descriptions such as (EPCglobal, 2010a; Främling, 2002; Huvio et al., 2002). From the table and above illustration, we can clearly see that most of these earlier works aimed to build a basic lookup service system that can only offer services R1, R2 and R3, but not the advanced services (R4 and R5). Additionally, from the performance perspective, the inquiry accuracy (R6), availability and reliability (R7), EPCDS independence (R8), and network extensibility (R9) are rarely discussed. Although the P2P-based designs have the potential to offer better scalability (R10), it still needs to be improved. Apparently, the existing research left much room for future EPCDS designs to fill. Our study plans to fill as many gaps as possible.

Table 3.1 Existing Studies for EPCDS Design

<table>
<thead>
<tr>
<th>Reference</th>
<th>Design Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R1</td>
</tr>
<tr>
<td>EPCONS (EPCglobal, 2008)</td>
<td>X</td>
</tr>
<tr>
<td>BRIDGE (BRIDGE, 2007a)</td>
<td>X</td>
</tr>
<tr>
<td>SLP (Afilias, 2008)</td>
<td></td>
</tr>
<tr>
<td>SLS (Polytarchos et al., 2008)</td>
<td></td>
</tr>
<tr>
<td>BGP (Khair et al., 2014a; Khair et al., 2014b)</td>
<td>X</td>
</tr>
<tr>
<td>ADS (Muller et al., 2010)</td>
<td>X</td>
</tr>
<tr>
<td>P2PONS (Fabian, 2008)</td>
<td>X</td>
</tr>
<tr>
<td>P2PONS (Xu et al., 2011; Dahbi et al., 2013)</td>
<td>X</td>
</tr>
<tr>
<td>DHTDS (Paganelli and Parlanti, 2012)</td>
<td>X</td>
</tr>
<tr>
<td>Hybrid DS (Liu et al., 2014)</td>
<td>X</td>
</tr>
<tr>
<td>RFID Resolution (Shrestha et al., 2010)</td>
<td>X</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>13</td>
</tr>
</tbody>
</table>
3.3.2. Security and Privacy

Although the security issues have been recognized as a significant part of EPCglobal, there are still very few works that have been done in this specific domain. We identified several studies that are closely related to this topic.

Grummt and Müller (2008) proposed a fine-grained access control for EPCIS, which is a rule-based context-aware policy language for describing access rights on a large set of EPCIS events. Although it provides an attribute level restriction, the paper focused on the security of a local EPCIS database only. There is no discussion for networking level security enforcement, and it is not feasible to define security policy for each individual client. Cantero et al. (2010) described a detailed design based on BRIDGE Project (BRIDGE, 2007a) models to offer more security protection. However, it concentrates on designing an authentication method, which employs X.509 and SOAP message, but offers no authorization model. Worapot et al. (2010) introduced and implemented a design of EPCDS with confidentiality for multiple data owners. Their approach built on the centralized DS and focused on the confidentiality in both data communication and application layers. They defined some very simple authorization rules, and it is not context-aware, nor does it apply to a large scale network due to the assumption of centralized DS. Also, they did not describe how to identify relationships between different entities or how to achieve anonymity. SHARIDS (Fabian et al. 2012) is a privacy enhanced discovery service for RFID-based product information. It provides protection for client side privacy by hashing the requested EPC cryptographically, but there is no solution addressed for the security of the information resource side. An encryption enforced access control model has been proposed by Kerschbaum and Chaves (2012). It presents an encryption scheme for exchanging item-level data in a distributed data repository. This work only uses the encryption method to ensure the security. It works at the database level, but not the network level. It neither
described how to identify relationships between different parties nor how to achieve anonymity. In fact, it is not an authorization model. SecTTS (Shi et al., 2012a) is a secure track and trace system for RFID-enabled supply chains. It is based on BRIDGE’s relay DS model, which allows EPCIS to publish relay policies to DS. Access control is implemented at the EPCDS side only, and relay policy is used to determine whether a query is permitted to be relayed to a company’s EPCIS or not. In this scenario, the relay policy only relates to each individual EPC code, which means the access control decision depends on the identity of the object, but not the identity of the subject, despite the fact that both object and subject will influence the final decision. Moreover, it is not a context-aware system, and cannot achieve anonymity. A secure and efficient EPCDS system (Shi et al., 2012b) is conceived and implemented for the local EPCIS database. This study discussed three types of relationships: the whole stream, the upstream, and the downstream relationships. However, it only considered the subject as an organizational client that has a business relationship with the target information resource, whereas there are more possible relationships in the real business world that need to be covered, e.g., the subject is an individual consumer or an organizational client who is not doing any business with the target information resource. In addition, it did not discuss network level security enforcement, or how to achieve anonymity. Another Distributed Hash Table (DHT) based Object Naming Service (ONS) model was proposed by Dahbi et al. (2013). This model used the public key infrastructure to encrypt the data and EPC code. Although it provides authentication and protects data integrity, it is essentially an ONS system and thus cannot fulfill the requirements of EPCDS. Kywe et al. (2013) introduced a security model to prevent the EPC event injection attacks under the EPCDS domain. This work only discusses an authentication model for each EPCDS participant to prove their ownership, whereas the holistic access control plan is missing. Shi et al. (2013) designed an authorization model for the EPCglobal network. This model works based on the three types of
security policies that are defined beyond the business relationships. However, it is implicit how EPCDS figures out the relationship between two specific entities in an automatic way. Moreover, neither the network level security enforcement nor the solution for interdependent security threat are missing. Another security scheme was proposed by Dahbi et al. (2014) to detect risky queries directed to an EPCDS. This model considered both the business habits of the stakeholders and the habits of the attackers, and used the Hidden Markov Model (HMM) to train and infer the risky queries. This is a good angle to detect the EPCDS security threats, however, it only addresses how to discover rather than how to stop them. It lacks a sufficient access control mechanism.

Most recently, the AnonEPC was described by Torvekar and Li (2014), which integrates with the SecDS model and provides a mechanism to protect the users’ privacy by allowing them to access the EPC event data anonymously; however, it leaves the privacy of the information resources unprotected.

Table 3. 2 summarizes and compares these studies, according to the seven key security requirements (R11-R17) that we identified in the previous section. As shown, none of these studies meet all requirements. Among the 12 studies, four studies (Cantero et al., 2010; Fabian et al., 2012; Kerschbaum and Chaves, 2012; Dahbi et al., 2013) did not meet any of the requirements; three studies (Jakkhupan et al., 2010; Shi et al., 2012a; Kywe et al., 2013) meet only one requirement; one study (Grummt and Müller, 2008) meets two requirements; three studies (Shi et al., 2012b; Shi et al., 2013; Dahbi et al., 2014) meet three requirements; and only one study (Torvekar and Li, 2014) meets four requirements. Clearly, none of these studies meet requirements two (anonymous), three (key publishing security) and seven (interdependent security), and thus put the EPCDS in hazardous position.
Table 3.2 Existing Studies for EPCglobal Security

<table>
<thead>
<tr>
<th>Studies</th>
<th>R11</th>
<th>R12</th>
<th>R13</th>
<th>R14</th>
<th>R15</th>
<th>R16</th>
<th>R17</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grummt and Müller, 2008</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>No security policy for each individual client.</td>
</tr>
<tr>
<td>Cantero et al., 2010</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Did not offer authorization.</td>
</tr>
<tr>
<td>Jakkhupan et al., 2010</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Very simple security design.</td>
</tr>
<tr>
<td>Fabian et al., 2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Protect client side privacy, instead of IR side.</td>
</tr>
<tr>
<td>Kerschbaum and Chaves, 2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Encryption scheme, no AC model.</td>
</tr>
<tr>
<td>Shi et al., 2012a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>AC did not consider clients’ relationship.</td>
</tr>
<tr>
<td>Shi et al., 2012b</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Did not explain how to identify relationships</td>
</tr>
<tr>
<td>Dahbi et al., 2013</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Essentially an ONS system, cannot fulfil the requirements of EPCDS.</td>
</tr>
<tr>
<td>Kywe et al., 2013</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Only resolving the EPC event injection attacks.</td>
</tr>
<tr>
<td>Shi et al., 2013</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Not explicit on how to identify relationships automatically.</td>
</tr>
<tr>
<td>Dahbi et al., 2014</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>It is a threat detection model, no access control mechanism.</td>
</tr>
<tr>
<td>Torvekar and Li, 2014</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>Protect the user privacy, but not information resources.</td>
</tr>
<tr>
<td>Sub-total</td>
<td>Yes</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>3</td>
<td>5</td>
<td>0 R12, R13 and R17 were not met by any existing study</td>
</tr>
</tbody>
</table>
3.4. Discussion and Implications

3.4.1. Research Direction

Based on the above discussion and analyses, we identify three research directions for consideration, and discuss potential challenges that we may face in the future design and implementation of EPCDS.

- **Research Direction 1:** Developing a discovery service model that can provide more advanced functionalities and services. Most existing DS solutions, including the DHT-P2P model, only provide the basic services, and thus need to be enhanced to provide more advanced services. These services will benefit all the individual entities, as well as the entire supply network. The first two architectures from the BRIDGE project and the ADS model can offer better services as compared with other models. Specifically, they provide us with a detailed description for their designs. The future design can take the advantage of these models, especially the ADS model to enhance the functionality.

- **Research Direction 2:** Designing a discovery service model that can improve the network performance, which includes the query accuracy, scalability, availability and reliability, independence, and extensibility. Compared to other architectures, P2P technology would be a better method for global-scale discovery services topology. It can facilitate the network to improve the availability and reliability, as well as the independence. In addition, the P2P network is much easier to extend, due to its flexibility. Considering these advantages, we believe that the P2P infrastructure would be a better underlay foundation to build the EPCDS architecture. However, the coordination of multiple discovery services needs to be managed.
• Research Direction 3: Designing a comprehensive security enforcement for the EPC discovery service. EPCDS needs to maintain the historical and serial level information within the network, as well as the EPC event data that are usually business sensitive and confidential. Anonymous authentication is the prerequisite of other security enforcement, though it may cause key publishing security issues, which need to be handled by the publishing control mechanism. In addition, the interdependent security issues are a serious threat, which must be resolved by the relationship-based federated access control mechanism.

3.5. Conclusion

In this chapter, we analyze the nature of EPCDS and its differences with ONS. We also explain the need for the discovery services from three major aspects: functionality and service, network performance, and security and privacy. For important indicators, we systematically summarize, classify and extend the EPCDS design requirements from the previous one. As we discussed, the EPCDS should provide more advanced services, rather than merely offer the basic services. Additionally, we provide four scenarios to exemplify our ideas.

We selected several typical DS models—the Directory Service model, Query Relay model, Aggregate Discovery Model, Service Location Protocol model, Service Lookup Service model, and P2P based model—for detailed discussion. Our comparative results show that the first three models can offer more complex and powerful services, but none of them are able to provide the AS, especially the ADS model that enforces the strengths of the other two, and ensures the inquiry accuracy. Except for the P2P-based models, most of the models can hardly handle the networking related issues, such as availability and reliability, independence, extensibility, and
scalability. P2P-based models show the potential for improving those specific performances. From the security and privacy aspect, a separate group of studies that concentrate on the security issues has also been summarized and compared against the security requirements.

Apparently, the peer-to-peer technology is a better underlay infrastructure to build the future EPCDS network. However, the design should also learn and include features from other models to provide more advanced services. An enhanced ADS approach should especially be integrated, in order to provide more advanced services and ensure the result accuracy as well as confidentialities. This work is distinct from other works in several aspects. First, more comprehensive EPCDS design requirements are categorized and extended from earlier works. Second, the most recent EPCDS models have been included and discussed. In addition, based on the requirements, we analyze and compare each model in detail. Next, the conclusion and future research directions are pointed out explicitly, which aim to guide the EPCDS design in the following chapters.
Chapter 4: Peer-to-Peer EPC Discovery Service Network Design

4.1 Introduction

RFID is one of the key enabling technologies used for automatic object identification, data capture and tracking in IoT applications. It is designed aiming to facilitate information exchange and the management of RFID-based EPCglobal network, which is an Internet based system offers interoperability under the complex multi-party environment. The greatest promise of the EPCglobal network is to offer the seamless sharing of data and item-level visibility for all the participants across the supply network, and to eventually achieve the blueprint of the IoT that every participant could easily share and access valuable business data in a secured manner.

Along with the fast growing of the IoT, a huge volume of data from the real-world objects and events is available on line to support the decision making. Therefore, a highly efficient and effective discovery service (DS) is the prerequisite, which is expected to eliminate the limitations of the EPCglobal development, such as high cost, lack of standardization, the security and privacy issues, the information sharing issues, the voluminous data issues, and the scalability issues (Michael and McCathie, 2005; Juels, 2006).

EPCglobal provides specifications but does not include detailed information regarding EPCDS architecture, interface and security. The positive aspect is that it gives users the flexibilities to design the EPC lookup system in different ways, following the core design principles. Besides the basic lookup function, the EPCDS should provide various extra services to enhance the functionality of the whole supply chain. On another aspect, EPCglobal network is a global scaled service that involves vast volumes of data, which is stored in distributed and heterogeneous databases. EPCDS must be able to fit into this situation by offering better network performance. Moreover, the security issue is always a concern for EPCDS design.
While several design prototypes have been proposed, most of them mainly focus on basic lookup and publishing services, scalability and traditional security issues, but overlooked advanced services such as standing query and message pushing services, core network performance requirements such as result accuracy, network independency and extensibility, and the security issues that specific to EPCDS such as the key publishing and access control issues. In this study, we propose to build a more sophisticate EPCDS network, based upon a peer-to-peer enabled framework, to enable multi-functionality, support better network performances and offer mutual anonymous authentication and key publishing security control. We identified 13 design requirements from an extensive literature in next section. Our ultimate goal is to satisfy all these design requirements.

The remaining part of the chapter is organized as follows: section 4.2 introduces the proposed P2P-enabled EPCDS model. In section 4.3, we discuss the experiment and computational results of the prototype. Finally, we conclude the study in section 4.4.

4.2. Proposed EPC Discovery Service Design

The fundamental goals of EPCDS are three-fold: (1) to provide an effective mean to find all the current and historical locations (EPCIS addresses) for a specific EPC key, (2) to provide a robust, scalable and efficient network infrastructure, and (3) to secure the operation procedures.

Figure 4.1 depicts the framework of the proposed design that composes four layers: Data/Physical Layer, Security Layer, Network Layer, and Application Layer.

The data and physical layer presents the information resources and the clients. The information resources include the organizational entities who share EPC event data. These data are captured and cleaned by each organization’s RFID readers and middleware ALE, and then
uploaded to the local EPCIS repository. The client represents either individual or organizational entity who consumes those services. Most of the time, a network participant plays both resource and client roles, since they provide information to other clients, and parallel requests service from the network.

![Figure 4.1 Proposed EPCDS framework](image)

The Security Layer protects all the network participants and secure the upper layer applications by interfering every transaction. This layer contains three components, the authentication mechanism, which is the foundation that supports the other two, the authorization and key publishing security control mechanisms. The security and privacy resolution are addressed in details in the next chapter.
All the EPCDS nodes connect together to construct the Network Layer, who receives and processes the requests from the providers and clients. The core function, key assignment function, leverages the P2P-based DS network infrastructure to distribute/retrieve each EPC key into/from different DS nodes. It enables the base services, publishing and lookup, that reside in the application layer.

The Application Layer consists of three components, Publishing Service, Lookup Service, and Advanced Service Module. Publishing Service takes care of the publishing requests that initiated by the service providers, while Lookup Service handles the lookup requests that come from the clients. Both services rely on the DS Table to locate the right DS nodes, and then publish/lookup the keys into/from the corresponding Item Table. They are the core services of the system, who enable the upper level application, i.e. Advanced Service Module.

We present and discuss our design in details from following aspects, network topology and infrastructure, key assignment function, publishing and lookup services, and advanced service below.

### 4.2.1. Network Topology and Infrastructure

Apparently, the P2P-based network infrastructure shows great potentials to fulfill the design requirements. The decentralized solution offered by P2P framework can avoid single point failure. It also demonstrates the possibility of improving the independency, extensibility and scalability. Therefore, we adopt a P2P-based infrastructure to combine with the advantages of non-P2P based models such as the result aggregation approach. Compare with other models, scalability is one of the major advantages of this novel P2P-based EPCDS model.
The proposed network infrastructure consists of numbers of peer nodes, where each node represents one EPCDS server that is composed of four major components, the Item Table and DS Table, Key Assignment function, Service Modules (Publishing, Lookup and Advanced) and Security Module. Figure 4.2 shows the EPCDS overlay network. The nodes at upper layer illustrates the unstructured physical machines, which are logically arranged into a circular linked underlying overlay network.

![Figure 4.2 EPCDS Network Topology and Infrastructure](image)

Before querying the EPC event data, the EPC key needs to be published into the DS network by employing the key assignment function that derived from consistent hashing method of Chord to locate a corresponding DS node for each EPC key. When the key assignment function is invoked, it matches the EPC key with a specific DS node according to the key value. The key Publishing Service will then distribute the keys into the right EPCDS nodes. Once the key is published, the user is able to query the relevant information via the EPCDS Lookup Service that is enabled by the same function. This P2P-based EPCDS network is able to fulfil the scalability requirements, and also shows the potential to provide better availability and reliability, extensibility and independency.
4.2.2. Key Assignment Function

The EPC key assignment is the principal function that exploits the EPC encoding schema to support all the EPCDS functionalities and services. The EPC Tag Data Standard (TDS) (EPCglobal, 2010b) defines several EPC Uniform Resource Identifier (URI) syntax schema, such as Serialized Global Trade Item Number (SGTIN), Serial Shipping Container Code (SSCC), and General Identifier (GID). In this study, we use the SGTIN encoding schema as an example to demonstrate the key assignment function. An example of SGTIN in decimal format is given below:

- General syntax: `urn:epc:id:sgtin-96:filter.PV.CPN.ItemRef.SerialNumber`
- Example decimal EPC URI: `urn:epc:id:sgtin-96:1.5.0614141.112345.400`

This scheme contains six fields. The first filed is the header that contains 8 binary bits to indicate the coding schema used. In this header, the coding schema is SGTIN-96, which indicates that the total length of the key is 96 binary bits. The second domain is the filter value, which uses a 3 bits of binary code, i.e. the decimal value space is [0, 7], to designate the packaging level. Behind that is a 3 bits partition value (PV) field, which determines the number of bits that used by the company prefix number (CPN). The forth field is CPN that is used to distinguish different manufactures. The length of CPN may vary from 20 to 40 bits of binary codes. In this example, the CPN is converted into a decimal number "0614141" that denotes the code of the manufacturer of this product. If two EPC keys carry the identical CPN, it means that they are manufactured by the same company. Next field is item reference domain whose length is dependent on the length of the CPN. The total length of fields four and five is 44 bits, which means the item reference ranges from 4 to 24 bits in length. Serial number field is placed at the tail of the code, which takes 38 bits. It is unique within each object class and is used to distinguish different items.
The hashing function used to generate the node ID for each EPC key is shown in formula (1):

\[ NID = CPN \% x^n \]  

Where, NID is node ID; \% is a modulo operation; x is a pre-decided index number that determines the amount of nodes; and n is the number of decimal digits of the company prefix number.

Based on Formula (1), if all the possible nodes are fully deployed, the amount of DS node will be \( x^n \), and the ID space ranges from 0 to \( x^n - 1 \), i.e. \( NID \in [0, x^n - 1] \). Normally, \( x \) can be set as 2, whereas in the extreme cases that need more number of nodes, \( x \) can be increased to a larger number, such as 3. When the key assignment function assigns an arbitrary EPC key, it applies Formula (1) to calculate the target node’s NID. This implies that the EPC keys will be distributed into the same node, while they have identical CPN because the remainders of CPN divided by \( x^n \) are equal.

Figure 4.3 exemplifies a simple EPC key assignment scenario. In this example, we assume that the company prefix number is 3 digits, which means \( n=3 \), and set the pre-decided index number \( x=2 \). If all the possible nodes are fully deployed, there will be \( x^n (2^3=8) \) nodes participant in this network, and their ID space is from 0 to \( x^n - 1 \), i.e. 23-1=7, which means \( NID \in [0, 7] \). In this EPCDS network, eight DS nodes that physically deployed in the actual network construct a circular logical overlay network. If there are four EPC keys need to be assigned, and their company prefix number are four different integers, such as 002, 014, 065 and 154. Formula (1) will be applied as following:

\[ CPN = 002, NID = CPN \% x^n = 002\%2^3 = 2 \]
\[ CPN = 014, NID = CPN \% x^n = 014\%2^3 = 6 \]
\[ CPN = 065, NID = CPN \% x^n = 065\%2^3 = 1 \]
\[ CPN = 154, NID = CPN \% x^n = 154\%2^3 = 2 \]
The result indicates that, any EPC key contains CPN 002, 014, 065 or 154 will be assigned to the corresponding DS node N2, N6, N1 or N2 respectively. Note that, despite that 002 and 154 have different company prefix number, they are assigned to the same DS node, N2, because after applying Formula (1) they will get the same reminder.

![Figure 4.3 An Example of 3 Digits CPN Key Assignment](image)

From the example we can learn that, if CPN is 3 digits in length, CPN ∈ [0, 999], when applying Formula (1) to assign all these CPNs, each node will receive $1000/23=125$ different CPNs. Each CPN associates with a number of EPC keys, for example 1000 in average. Then, each node needs to maintain $1.25E+5$ EPC keys. When the number of EPC key is large enough, the key distribution should be roughly even for each DS node, though the number of EPC keys that associate with each CPN may vary. Hence, by employing this consistent hashing function, a large amount of EPC key can be distributed evenly into a small number of DS nodes, which amount is determined by parameters x and n. When there are more keys, the number of nodes could be increased accordingly.

The main strength of this key assignment function is the scalability, which is the most important requirement for EPCDS network design. With the keys evenly distributing into each node, it enables the global scale usage of EPCDS network.
4.2.3. Publishing Service

The EPCDS Item Table and DS Table are the prerequisite of the publishing and lookup services, and therefore they deserve detailed discussion beforehand. Both of them are maintained by each independent EPCDS node. Item Table stores information relates to each EPC key, while DS Table maintains addresses of all the DS nodes.

Item Table can reflect the movement path of each product, because it stores the IP address of each EPCIS that hosts the EPC event data of this product. It contains two types of attributes, EPC key (EPC) and position ($P_i$). The first attribute stores the EPC key that serves as prime key of the table. The other attribute “$P_i$” is used to store the IP addresses of each EPCIS that possesses the information about this product, where “$i$” is a positive integer that indicates the relative position of each organization (EPCIS) that locates in the supply chain. The value format of “$P_i$” is <IS.name:IP>.

Table 4.1 shows an example of the EPCDS Item Table, where each row represents one supply chain for a specific product. In this dissertation, if an EPCIS repository belongs to a company called A, the repository will be denoted as IS.a. When a user requests data from company A, it needs to access the data in IS.a, which can be located by the $P_i$ value <IS.a: 47.92.101.23> that indicates the IP address of company A’s EPCIS repository is 47.92.101.23. The Item Table can also imply the supply chain partners and their relationships. For example, if the product, EPC e123, was produced by company A, and then moved to distributor B, and finally purchased by company C. These three companies would report EPC e123 to the EPCDS serially, which is stored by row 1 (e123) of Table 4.1. In this way, the Item Table reflects the movement path for each product, such as P1 must be the upstream company of P2, while P3 must be the downstream company of P2. This is based on the rationale that one object cannot physically present at more than one location synchronically. Once the company receives the object, the EPC
key will be reported to the EPCDS instantly, which is automatically accomplished by the RFID system and the key Publishing Service offered by EPCDS. This innovative Item Table design provides a solid evidence to the system to identify the relationship among entities.

Table 4.1 Example of EPCDS Item Table

<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
</tr>
</thead>
<tbody>
<tr>
<td>e123</td>
<td>IS.a: 47.92.101.23</td>
<td>IS.b: 120.22.1.17</td>
<td>IS.c: 23.62.18.28</td>
</tr>
<tr>
<td>e243</td>
<td>IS.d: 58.2.211.156</td>
<td>IS.e: 40.41.51.127</td>
<td></td>
</tr>
</tbody>
</table>

DS Table is used to store the locations of the EPCDS nodes within the network. The table contains two attributes, the node ID (NID), servicing as the prime key, and “addr.” stores the IP addresses of all the EPCDS nodes. Table 4.3 gives a simple example of the DS Table, for instance, it indicates that the DS0’s IP is 124.72.54.12 and DS4’s IP is 218.207.217.233. DS Table enables the EPCDS nodes to find and collaborate with other peers. Every EPCDS node needs to keep one copy of the DS Table.

EPCDS leverages a two-layer decentralized Item Table and DS Table to enhance Availability and Reliability (R7). In the first layer, all the EPC keys are divided into a number of segments and distributed into the corresponding DS nodes (Item Tables). This design prevents the single point failure issue, since there is no single DS node can affect the function of other nodes. Moreover, the valuable EPC event data are spread into a large amount of EPCIS repositories, which further decentralized the services into another layer. Under this circumstance, one failure entity will not be able to harm the functionality of the entire network infrastructure. In addition, a failure recovery function can be easily accomplished by establishing the Item Table backup mechanism. For example, it requires each EPCDS node to duplicate a copy of the Item Table of its next DS node, whose ID is \((NID+1)\%x^n\).
However, there is no need to maintain redundant DS Tables, since every DS node keeps a complete DS Table that stores addresses for all the nodes. If one DS node is down, it can be substituted by any other node. In addition, while we are using the 96 binary bits coding schema, which declares that the company prefix number is 20 to 40 binary bits, that means there are $2^{20}$ to $2^{40}$ possible CPN. Covert to decimal, it is approximately to $10^6$ to $10^{12}$ bits. According to the key assignment function, the number of DS nodes will be $2^6$ to $2^{12}$, i.e. $[64, 4096]$. It proofs the feasibility that even in the worst case, it requires few resources (4096 entries) to maintain a complete DS Table.

Figure 4.4 shows the Publishing Service architecture that enables the typical key publishing scenario. The key Publishing Service highly relies on the DS Table and Item Table design. When a product arrives the loading deck of a supply chain participant (EPCIS), the RFID reader automatically scans the RFID tag attached on this product, and extracts the EPC key. Once done, the system will do two things. First, it updates the local EPCIS repository through the middleware server ALE. And in the meanwhile, it invokes the EPCDS Publishing Service who uses the key assignment function to figure out which destination DS node it should report to. After obtaining the node ID, the DS Table is able to return the address of the destination node. The Publishing Service can now connect with the right destination DS node, and publish the EPC
key to its Item Table. This procedure is completed by sending a two tuple message \(<EPC \text{ Key, IS:IP.address}>\), which tells the EPCDS that this EPCIS possesses this EPC key.

When destination DS node receives these two parameters, it looks up the first value, the EPC key, in its Item Table. If there is no match for this key, it means this is the very first time that this EPC appears in the entire EPCDS network due to it is the only DS node that is identifiable by this key. Thus, the DS node will create a new row for this key in its Item Table. The first parameter of the message (EPC key) will be inserted into the EPC column, and the second parameter (IS:IP.address) will be placed at \(P1\) that represents the first position of this supply chain, because it is the first organizational entity that register this EPC key.

However, if DS node found one match EPC key in the Item Table, it will insert the second value, IS:IP.address, to the last position of the \(Pi\) domain, instead of creating a new row. For instance, if a new EPCIS, IS.f:46.3.134.5, reports to EPCDS to claim that it possesses EPC e123 and e243, it will send two messages, \(<e123, \text{IS.f: 46.3.134.5}>\) and \(<e243, \text{IS.f: 46.3.134.5}>,\)
to the EPCDS. The DS node will insert the parameter “IS.f.6.6.6.6” to P4 and P3 of the e123 row and the e243 row, refer to Table 4.1.

This key Publishing Service function allows EPCIS to report its maintained EPC keys to the right destination DS node, by applying Formula (1) to obtain the DS node ID. The major advantages of this approach include that a specific EPC key will be maintained in only one corresponding EPCDS Item Table, which keeps a complete records for this key. The key publishing procedure is a one to one mapping process that is able to fulfil the Publishing (R2) and Multi-Publishing (R3) requirements. Once complete this, clients can query to these public data through the EPCDS Lookup Service.

4.2.4. Lookup Service

After EPCISs publishing the EPC keys to each DS node, those EPC keys will be distributed into different Item Tables. When clients want to query information about certain EPC keys, they send their requests via EPCDS Lookup Service, which enables them to connect with an adjacent DS server. The DS server will then apply key assignment function to these EPC keys and calculate which EPCDS nodes (NID) are responsible for hosting these records. According to the DS Table, the lookup service is able to find the destination DS node, and extract addresses of the target EPCIS repositories from the Item Table. Instead of asking the clients to query into these information resources directly, the proposed Lookup Service adopts the result aggregation method that is derived from Aggregate Discovery Service (ADS) model, which collects and aggregate all the sub-results from each EPCIS repository, and thereafter present the final result view to the client. Figure 4.5 shows the Lookup Service architecture.
In this lookup process, two possible cases may occur, first is the destination DS node is the same as the requested node, and the other case is the destination node is different from the current requested one.

The first case is simple due to the requested EPC key just happens to be maintained by itself, and therefore it just needs to look up into its own Item Table, and find out the records of this EPC key. And then the result aggregation method will be applied.

![Figure 4.5 Lookup Service Architecture](image)

The other case is more complicated, since it involves multi-EPCDS cooperation. After obtaining the NID of the destination DS node, the requested node needs to look up the DS Table to locate the IP address for the destination DS node, and then redirect the query to that node. The destination DS node will search its Item Table and find out the right entry of the target EPC key, and then apply the result aggregation method. Figure 4.6 exemplifies the query process of the newly proposed EPCDS Lookup Service. See for example,

a) EPCDS client CLT wants to know the details information about EPC e123, thus CLT initiates a query request for this EPC through the EPCDS query interface. The request is sent to an adjacent DS server, i.e. DS0.
b) When DS0 receives the query request, it extracts the CPN from the EPC key, and then apply Formula (1) to calculate the destination DS node ID. Let assume \( \text{NID} = \text{CPN} \mod x^n = 4 \).

c) Because the node ID of the destination DS is 4, which means the target EPC information is held by DS4, the requested node DS0 needs to search the IP address of DS4 in its DS Table.

d) Once obtaining the IP address of the destination node DS4, the requested node DS0 is able to communicate with it, and redirect the query request to DS4.

e) When DS4 receives the query request for EPC e123, it will look for this key in its Item Table. After DS4 finds the right entry, it obtains the IP addresses of all the EPCIS that have possessed information about EPC e123. In this example, DS4 gets the IP addresses of IS.a, IS.b and IS.c, which are 47.92.101.23, 120.22.1.17 and 23.62.18.28 respectively.

f) Now, DS4 is able to conduct the result aggregation method that queries to IS.a, IS.b and IS.c. Each operation is independent to others, and these target EPCIS will respond by detailed EPC event data of e123 that is stored in their EPCIS repository. Thus, three sub-results will be collected by DS4 separately. They are a.e123, b.e123 and c.e123.

g) When DS4 receives all the sub-results, it needs to filter and combine these data. The three sub-results will be aggregated at DS4 to a complete final result. In this case, it is the union of the sub-results, \( \{a.e123 \cup b.e123 \cup c.e123\} \).

h) The aggregated final result is sent back to the requested node DS0 by the destination node DS4.

i) DS0 presents the final result to the client CLT without any change.
In this process, EPCDS only sends client’s query request to the target EPCIS for query, which excludes client’s identity information. After gathering the sub-results, the EPCDS will filter and organize them to the final result, which will be presented to the client without any identity information about the information provider, i.e. EPCIS. Thus, prevent from disclosing individual identity to the unknown entities that might be malicious.

As can be seen that this mechanism achieves the Lookup Service (R1), which is the foundation of other advanced services. This highly efficient and effective service can improve the performance of the entire network by quickly locating the target resources, because the EPC key and EPCDS node mapping is a one to one process that a specific key is registered at nowhere but the corresponding destination DS node. This design enables the result aggregation method.
Because the complete target EPCIS list is managed by the Item Table of the destination DS node, it knows how many sub-results are expected. If any EPCIS did not respond properly, the DS node is able to notice and resend the request to that information provider. This will ensure the final result is complete and correct. In case the EPCIS cannot respond permanently due to the server is down or other force majeure factors, the DS node can point out which EPCIS is missing from the network, and thus DS can include this exception in the final result view that is going to be presented to the client. As a result, the proposed Lookup Service demonstrates the effectiveness that guarantee the accuracy of the final result (R6).

4.2.5. Advanced Services

The Advanced Services include the subscription/standing query service, and the pushing service. Although they share some similar features, the standing query service is initiated by the clients while the pushing service is triggered by the information providers (EPCISs). They can be integrated with EPCDS network, and enabled by Advanced Service Module as shown in Figure 4.7, which includes four components: Subscription Interface, Schedule Checker, Event Manager and Message Manager.

When client initiates the subscription request for a specific EPC key via the Subscription Interface, it will redirect user to the Message Manager (MM) for registration. Client needs to input its contact information, e.g. email address, into MM’s client information repository. In the meanwhile, Subscription Interface triggers the Lookup Service to locate the destination DS node for the subscribed EPC, and places the standing query to the corresponding node.

The standing query can be divided into two categories, the time based standing query and the event triggered standing query. The Schedule Checker takes care of the time based
subscriptions. When a time based standing query is placed by the client, the Lookup Service facilitates the subscription interface to locate the destination DS node, where the subscribed EPC key is inserted into the Schedule Checker for record. The Schedule Checker probes the subscribed key periodically, and uses the result aggregate method to obtain the updated EPC event data. When all the new updates are aggregated and returned to the Schedule Checker, it needs to forward these messages to the Message Manager, who maintains a list of the subscribers and notifies the client via email.

The event triggered standing query is handled by the Event Manager. When such query is placed by the client, the Lookup Service is invoked to locate the destination DS node. The subscribed key will be registered to the Event Manager of that node, who manages the event triggered subscriptions. According to the Item Table, the Event Manager marks these keys as subscribed in the corresponding EPCIS repositories. Hence, these EPCISs know which EPC keys are registered for the event triggered subscription. When the EPC event data of the subscribed
keys are updated, the EPCIS sends the updates back to Event Manager. The Event Manager forwards these messages to the Message Manager, who is responsible for delivering the updated results to the right subscribers.

Pushing Service is initiated by the information resources, and then processed by the Event Manager of the destination DS node. When emergency occurs, for instance one company decides to recall a product, it can trigger the pushing service to notify all the relevant entities within the same supply network. The EPCIS reports this EPC key to the Event Manager, who is able to locate the target EPCISs that hosting this key, and thereafter push the notification to the hosts.

As can be seen that the Advanced Service Module is able to achieve Subscribing (R4) and Pushing Service (R5). The advantages of this method include that the services can be conducted in a highly efficient and scalable manor, because the foundation of these advanced services is the basic Lookup Service. In addition, the EPCDS network is independent to other components (R8), e.g. EPCIS, Subscription Interface and Message Manager, which means that if one component changed, it would not affect the functionality of the EPCDS network.

4.3. Analysis and Evaluation

4.3.1. Requirements Analysis

We explicitly point out 10 EPCDS design requirements that fall into Functionality and Service, and Network Performance groups. This section analyses our design against these requirements.
As aforementioned discussion, all the 5 requirements from the Functionality and Service group can be fulfilled by our proposed design. By applying Formula (1), the system will map an EPC key to one specific EPCDS node, who keeps a complete record for this corresponding key. This key assignment function enables a set of highly efficient and effective base services that include Lookup Service (R1), Publishing Service (R2) and Multi-Publishing Service (R3). Other than that, the Advanced Services, such as Subscribing Service (R4) and Pushing Service (R5), can be enabled by the Advanced Service Module, which is based on the Lookup Service.

The Network Performance group contains six requirements. The Result Accuracy (R6) is proved by result aggregation method. Because DS node keeps a complete EPCIS list for every corresponding EPC key, when querying a specific key, the destination node knows how many sub-results will be returned. DS node will resend the request to the information resource that fails to respond to ensure the final result is complete and accurate. In case any EPCIS server is down for some reason, the DS node is aware of which resource is missing, and thereafter it can notify the client. Availability and Reliability (R7) requirement is also satisfied by the Item Table and DS Table coordination. The two-layer decentralization design avoids the single failure issues, and the backup Item Table plan offers the failure recovery mechanism. Independency (R8) is another critical requirement. The proposed EPCDS network is independent to other components, e.g. EPCIS, Advanced Service Module and security component, which means if these components altered, the EPCDS do not need to make any change accordingly to keep functioning properly. This EPCDS infrastructure can also be extended (R9) by integrating with other components, especially the security countermeasures, such as our anonymous authentication and publishing security control mechanisms, that can be implemented at each DS node. Scalability (R10) is the most important design requirement of EPCDS. The newly designed network is born for the large scale usage due to the key assignment function is based on a highly efficient consistent hashing
function that enables the real world global scale environment usage by distributing an
decentralizing the EPC keys.

4.3.2. Performance Evaluation

Figure 4.8 illustrates the prototype architecture that is implemented, which contains seven
modules: key generator, DS interface, key assignment function, lookup and publishing service,
DS Table and Item Table.

The key generator mimics the behaviour of client or information provider to generate a
set of inputs randomly, and inputs them into the DS interface. Depends on different parameters of
the input, the DS interface specifies the request types. If the input contains only one parameter,
the EPC key, it is a lookup request, and thus the DS interface will invoke the lookup service module to proceed the Lookup Service. If the input contains a two tuple parameters, <EPC key, IS:IP.address>, the DS interface will invoke the publishing service module to conduct the Publishing Service. Once receiving the request, the service module triggers the key assignment function to calculate the NID of the destination DS node. The address of the Destination node is returned by the DS Table. According to the address, the service modules can either query or write the target message from or into the corresponding Item Table. Finally, the destination node returns the target EPCIS addresses to the service module and DS interface. The whole procedure is completed, when the DS interface print out the result view. The prototype evaluates the relative performances of publishing and the lookup services, who reflect the performance of the core function, key assignment function. The reason we select these two services for implementation is that they are the base services to support other advanced services, and determine the performance of the entire network function.

We implement all the techniques in Java programming language and framework under Java Virtual Machine version jre1.8.0_25. The experiments and evaluations were performed on a test bed with an Intel Quad Core I7-2600 CPU (3.40 GHz) and 16 GB RAM, running in Windows 7 64x Professional operation systems. The experimental design, performance metrics and test results are discussed in this section.

4.3.2.1. Experimental Design

We design five independent experiments for performance assessment. Each experiment includes three parameters: the amount of keys, length of EPC keys and the length of CPN, which are summarized in Table 4.3. In order to be statistically convincible, each experiment is repeated 20 times, and the EPC keys are generated by different seeds randomly.
The first experiment measures the distribution patent of the key assignment function. We run key assignment function to assign \{10, 102, 103, 104, 105\} keys, while keep the length of the EPC key and CPN at 10 and 4 digits constantly. The reason we use 10 and 4 digits length of EPC key and CPN is that it can simulate the real world situation. As we know the EPC key is 96 binary bits in length, and if we exclude some flag bits, the total length is around 80 bits, and the company prefix number is from 20 to 40 bits, which is at the similar the proportion of our assumption. Thus, the number of node is 16 (xn=24), NID is from 0 to 15. In addition, the tested number of keys range from small amount to large amount. The small amount means it is in the same orders of magnitude as the number of node, e.g. 10, while the large amount means it is several orders of magnitude greater than the number of node, i.e. 104 or 105. We collect the amount of keys that received by each node, and then calculate the average results of each test case, which are accurate to two decimal places. The chi-square test result is calculated by SPSS software to prove the distribution patent.

Table 4.3 Experiment Parameters

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Length of EPC</th>
<th>Length of CPN</th>
<th>Number of Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>4</td>
<td>10, 10², 10³, 10⁵, 10⁶</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>[1, 5]</td>
<td>50000</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>[1, 5]</td>
<td>50000</td>
</tr>
<tr>
<td>4</td>
<td>[5, 10]</td>
<td>4</td>
<td>50000</td>
</tr>
<tr>
<td>5</td>
<td>[5, 10]</td>
<td>4</td>
<td>50000</td>
</tr>
</tbody>
</table>

In experiment 2-5, we publish 50000 keys for each of the experiment. The reason is to simulate the real world scenario that the key amount is several orders of magnitude greater than the number of node.

Experiment 2 assesses the Publishing Service, when the length of CPN increases. We keep the length of EPC key at 5 digits constantly, while increase the CPN length from 1 to 5
digits, in increments of 1 digit per test case. We record the publishing time of each tests, and calculate the average results of these 20 rounds tests.

Experiment 3 evaluates the Lookup Service, when the length of CPN increases. We keep the length of EPC key at 5 digits constantly, while increasing the CPN length from 1 to 5 digits, in increments of 1 digit per test case. We record the lookup time and the path length of each tests, and calculate the average results of these 20 rounds tests.

Experiment 4 assesses the Publishing Service, when the length of EPC key increases. We keep the length of CPN at 4 digits constantly, while increasing the key length from 5 to 10 digits, in increments of 1 digit per test case. We record the publishing time of each tests, and calculate the average results of these 20 rounds tests.

Experiment 5 conducts the Lookup Service, when the length of EPC key increases. We keep the length of CPN at 4 digits constantly, while increase the key length from 5 to 10 digits, in increments of 1 digit per test case. We record the lookup time and path length of each tests, and calculate the average results of these 20 rounds tests.

4.3.2.2. Performance Metrics

The relative performance of the system is measured by four metrics, the load balance, the publishing time and lookup time, and the path length.

Load balancing is a method that gauges the distribution of work load into different computing resources, i.e. DS nodes. The main purpose is to enhance the scalability of the system. It is measured by how evenly the published keys are distributed among nodes. It tests the capability of the key assignment function. In the ideal cases, if there are $k$ keys and $n$ nodes, we would like to see that each node will be assigned approximate $k/n$ keys. The uniform distribution of the data can be proved by chi-square test, which shows in Formula (2), where $f_0$ is the actual observation frequencies and $f_e$ is the theoretical frequencies, i.e. $k/n$. 

87
\[ x^2 = \sum \frac{(f_0 - f_e)^2}{f_e} \sim x^2 \]  

(2)

The second and third metrics are publishing and lookup times, based upon which we can explore the impact of the length of EPC key and CPN on system performance. The publishing/lookup time is defined as the processing time for completing a Publishing/Lookup Service. The processing time is considered as the time costs since the DS interface receives the request until it prints out the final result. We expect to see that when the length of EPC key or CPN increase, the processing time remains stable or even decrease, which demonstrates better scalability.

The performance of the routing protocol shows strong connection with the path length between two arbitrary nodes (Stoica et al., 2001), the shorter path length usually means better network performance, i.e. scalability. In the EPCDS context, we calculate path length as the number of hops required to complete an end-to-end Lookup Service.

4.3.2.3. Results and Discussion

The result of the first experiment is summarized in Table 4.4, which lists the mean of the number of keys that are received by each DS node, NID ∈ [0, 15]. After conducting the chi-square test by applying Formula (2), the SPSS results in Table 4.5 show the significant level for all the five cases are greater than 0.05. Thus we can conclude that no matter how many keys are input to the system, from small to large amount, it demonstrates good load balance capability that the published keys are evenly distributed to each node, from N0 to N15. This is the fundamental characteristic to support a better network performance, such as the scalability, and availability and reliability.

The impacts of different length of keys and CPNs on the time complexity of the publishing and lookup services are measured by experiment 2-4. The results are illustrated in
Figure 4.9. The results (a) and (b) indicate that the length of CPN has significant influence on the time complexity of the key publishing and lookup services. When the size of EPCDS node increases, i.e. CPN length increase, the system processing time decreases accordingly. The reason is that when there are more DS nodes are deployed in the network, the workloads will be decentralized into different nodes, and thus increase the efficiency of the holistic system. The results prove that our new designed EPCDS network has great capability to handle large key space, and thus it is able to support large scale usage.

Table 4.4 Key Distribution in Each Node

<table>
<thead>
<tr>
<th>Total Key</th>
<th>10</th>
<th>100</th>
<th>1000</th>
<th>10000</th>
<th>100000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node0</td>
<td>0.65</td>
<td>6.1</td>
<td>62.25</td>
<td>625.15</td>
<td>6250.35</td>
</tr>
<tr>
<td>Node1</td>
<td>0.75</td>
<td>6.3</td>
<td>62.05</td>
<td>625.45</td>
<td>6250.2</td>
</tr>
<tr>
<td>Node2</td>
<td>0.55</td>
<td>6.25</td>
<td>62.6</td>
<td>625</td>
<td>6249.85</td>
</tr>
<tr>
<td>Node3</td>
<td>0.7</td>
<td>6.15</td>
<td>62.55</td>
<td>625.1</td>
<td>6249.7</td>
</tr>
<tr>
<td>Node4</td>
<td>0.65</td>
<td>6.4</td>
<td>62.45</td>
<td>625.1</td>
<td>6250.1</td>
</tr>
<tr>
<td>Node5</td>
<td>0.6</td>
<td>6.25</td>
<td>62.7</td>
<td>625.15</td>
<td>6250.3</td>
</tr>
<tr>
<td>Node6</td>
<td>0.7</td>
<td>6.15</td>
<td>62.85</td>
<td>624.5</td>
<td>6249.15</td>
</tr>
<tr>
<td>Node7</td>
<td>0.55</td>
<td>6.4</td>
<td>62.35</td>
<td>624.85</td>
<td>6250.7</td>
</tr>
<tr>
<td>Node8</td>
<td>0.6</td>
<td>6.3</td>
<td>62.7</td>
<td>624.65</td>
<td>6250.05</td>
</tr>
<tr>
<td>Node9</td>
<td>0.65</td>
<td>6.2</td>
<td>62.85</td>
<td>624.7</td>
<td>6249.45</td>
</tr>
<tr>
<td>Node10</td>
<td>0.55</td>
<td>6.35</td>
<td>62.35</td>
<td>625</td>
<td>6250.9</td>
</tr>
<tr>
<td>Node11</td>
<td>0.6</td>
<td>6.3</td>
<td>62.65</td>
<td>624.7</td>
<td>6249.25</td>
</tr>
<tr>
<td>Node12</td>
<td>0.6</td>
<td>6.2</td>
<td>62.65</td>
<td>625.25</td>
<td>6249.85</td>
</tr>
<tr>
<td>Node13</td>
<td>0.65</td>
<td>6.25</td>
<td>62.45</td>
<td>624.95</td>
<td>6250.2</td>
</tr>
<tr>
<td>Node14</td>
<td>0.65</td>
<td>6.25</td>
<td>62.1</td>
<td>625.4</td>
<td>6249.95</td>
</tr>
<tr>
<td>Node15</td>
<td>0.55</td>
<td>6.15</td>
<td>62.45</td>
<td>625.05</td>
<td>6250</td>
</tr>
</tbody>
</table>

The results (c) and (d) denote that when the length of CPN is fixed, the length of EPC key has very limited impact on the key publishing and lookup services. Each case takes a similar duration for publishing and querying same amount of EPC keys. This also proves that when the key space increase, the system can still work in an efficient manner.
From the results, we also notice that it takes more time to complete the Publishing Service than the Lookup Service. The reason is that the Publishing Service needs to write into the Item Table, while the Lookup Service merely extracts the data from the Item Table and returns.

Table 4.5 Load Balance Chi-Square Test Result

<table>
<thead>
<tr>
<th>Key Amount</th>
<th>Chi-Square</th>
<th>df</th>
<th>Asymp Sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>3.375a</td>
<td>4</td>
<td>0.497</td>
</tr>
<tr>
<td>10²</td>
<td>3.250a</td>
<td>6</td>
<td>0.777</td>
</tr>
<tr>
<td>10³</td>
<td>2.750a</td>
<td>9</td>
<td>0.973</td>
</tr>
<tr>
<td>10⁴</td>
<td>2.000a</td>
<td>11</td>
<td>0.998</td>
</tr>
<tr>
<td>10⁵</td>
<td>1.500a</td>
<td>13</td>
<td>1.000</td>
</tr>
</tbody>
</table>

The path length is calculated based on the prototype, where all the nodes are linked into a circle. If the DS node only knows the address of its successor, it becomes a circular linked list. In order to reach the target node, the request needs to move along the circle and eventually reach the desired peer. This may lead to the worst case that takes $O(n)$ steps to get the destination node, $n$ is the number of nodes. However, in our design, each node keeps one complete copy of DS table that includes the addresses for all the peers. According to the CPN of the target EPC key, the DS is able to calculate the node’s NID, which will point to one unique node who is hosting this key. In this manor, each request only needs one hop to reach the destination DS node, and then locates the EPCIS by searching the Item Table. At last, query into the resource as the second hop. Hence this approach needs two hops for reaching the target information resources, which means the path length is 2 and the Lookup Service is resolved in $O(2)$. We record the path length in experiments 2 and 4, and show the result in Figure 13 (a) and (b). According to the results, we can conclude that when the length of CPN and key increase, the path length remains constantly, which demonstrate good system scalability and efficiency.
4.5 Conclusion

In this chapter we identify 10 explicit EPCDS design requirements from functionality and service, and network performance aspects. A novel P2P-based model is proposed based on a
consistent hashing enabled key assignment function. It enables the network to distribute a huge amount of EPC keys into a small number of DS nodes, and therefore allows the users to publish and retrieve the target information efficiently and effectively. A fully deployed EPCDS network can satisfy the global scale usage expectation. Comparing with other existing EPCDS designs, this newly proposed model can offer more sophisticated services, enhance the network performances and secure the operation procedures.

A prototype is implemented for the proof of concept purpose. The experimental results show that our model has good load balance capability that can distribute large volume of keys into different DS nodes evenly. The time complexity of the operation is stable, when larger space of EPC key applies. When CPN grows larger, which implies there are more organization entities participant into the network, the number of EPCDS nodes will increase accordingly. It has been demonstrated by the experiment that the system works more efficiently, if greater number of nodes are deployed within the network.

The major contributions of this work include the followings. It does not only support the Basic Service, e.g. lookup and publishing services, but also the Advanced Service, e.g. subscribing and pushing services. It is able to fulfil the network design requirements, especially demonstrate great scalability that can fit into the real world situation. Other than that, the model guarantees additional strengths such as inquiry accuracy, availability and reliability, and also offers independency, extensibility and scalability. A unique mutual anonymous authentication mechanism is also described in this work, which protects the privacy of both service requestor and provider. To the best of our knowledge this is the first work that addresses the key publishing security threat in EPCglobal network, while its countermeasure is introduced as well. We believe that this will be a valuable complementary component for the EPCglobal and IoT network. The security issues will be discussed in next chapter.
Chapter 5: Relationship-based Federated Access Control Model for EPC Discovery Service

5.1. Introduction

EPCglobal network aims at developing a universal identification system with an open architecture to provide interoperability in a complex multi-party environment (Barchetti et al., 2011). In order to meet the demands of information sharing and retrieving, a key component, EPC Discovery Service (EPCDS), is required for the object discovering purpose. With a suitable service oriented architecture, the Electronic Product Code (EPC) can be used as a lookup key for locating information resources as well as extracting relevant data from these resources (BRIDGE, 2008).

In order to take the advantage of EPCglobal network, information from different organizations should and must be shared to create a smart and highly networked environment. Managing the EPCglobal network faces several major challenges: high cost, lack of standardization, security and privacy, information sharing control, voluminous data, and network scalability (Michael and McCathie, 2005; Juels, 2006). Since a vast volume of data is stored in widely distributed and heterogeneous databases under highly dynamic and diversity contexts making the task of securing this global scaled network quite challenging. Especially since the EPC event data are stored and managed by each organization’s local database independently, their security requirements, security structure, method, and policy are different; thus, it may result in a big security concern, known as the interdependent security threat (Kunreuther and Heal, 2003). Moreover, EPCDS is an open system that has no control over the key publishing procedure, it exposes to the key publishing security threat.
The EPCDS is expected to take care of these highly dynamic and diversity inquiries, and in the meantime to ensure the information security of all EPCDS players. In this chapter, we propose and test a relationship-based federated access control model to address the key publishing security issue and the interdependent security concern. To the best of our knowledge, this is the first study that identifies and examines in details the causes and impacts of these two threats in EPCDS. We have proposed a security countermeasure to secure the network at both network and database levels to fulfill the security requirements of each individual entity without compromising others’. This chapter is organized as follows: section 5.2 introduces the underlying EPCDS infrastructure that this security model built on and a comprehensive security analysis on the EPCglobal network. In section 5.3, the proposed relationship-based federated access control model is described. In section 5.4, we discuss the experiment and computational results of this model. Finally, we conclude the study in section 5.5 and suggest directions for future research.

5.2. EPCglobal Network Background and Security Analysis

5.2.1. EPCglobal Network Background

Several designs of EPCDS have been proposed, for example, the RFID Directory Service and RFID Query Relay models (BRIDGE, 2007a), Service Location Protocol (SLP) model (Afilias, 2008), Service Lookup Service (SLS) model (Polytarchos, et al., 2008), and Aggregating Discovery Service (ADS) model (Muller et al., 2010). Another group of designs were based on the Peer-to-Peer approach, such as DHT-P2P model (Fabian, 2008), P2PONS (Xu et al., 2011), P2P RFID resolution framework (Shrestha et al., 2010) and Distributed P2P DS (Lorenz et al., 2011). Most of these designs, however, were built upon a simple lookup service by establishing
connection between client and information resource, and thus they are limited in providing more advanced features. Additionally, the scalability, result accurateness, availability, reliability, independency and extensibility issues are rarely discussed by these works. Most important is that these models are lacking sufficient security countermeasures, which restricts their applications in practice.

![EPC Discovery Service Framework](image)

**Figure 5.1 EPC Discovery Service Framework**

Usually, the EPC Discovery Services is deployed in the middle layer with information resources and system users, where is also the right place to integrate the security enforcement. The EPCDS infrastructure we employed is based on Figure 4.6. While the object moves among network partners, the associated EPC event data are captured by different information resources and stored locally. As shown in Figure 5.1, EPCDS is sitting on the top of data repositories to manage the data retrieving, distributing and security functions. It simplifies the data exchange process by offering a service that links all the relevant information of a certain object across the whole supply network. It enables the client to locate all the entities that have possession of a given object and to share the relevant EPC event data, and therefore allows the network
participants to proactively manage their supply chains and ultimately gain further benefits that EPCglobal promised.

The information resource shares its valuable business data with other network partners. These data physically located in each organization’s EPCIS local repository, which is typically a database. In this dissertation, if an EPCIS repository belongs to a company A, it will denote by IS.a. When a user requests data from company A, it needs to access the data in IS.a. These data, also called EPC event data, are usually business sensitive for each individual organization. EPC event data may include but not limit to product description, inbound time, outbound time, upstream company, downstream company, buying price, selling price, product location, as well as other information, such as product quality, temperature, and humidity etc. An example of EPCIS database is shown in Table 5.1, which enumerates some basic attributes of EPC e123 and e243. It contains eight attributes, which are: C1 Product Description, C2 Inbound Time, C3 Outbound Time, C4 Upstream Company, C5 Downstream Company, C6 Buying Price, C7 Selling Price, and C8 Product Location. The fine-grained authorization model should enforce the access control at the attribute level, which means the security policy can be acted at each cell of the table. For instance, the security rules may only allow one client to access columns C1 and C3 of e123, while another client is granted to access columns C2, C5 and C8 of e123.

Table 5.1 Example EPCIS Database

<table>
<thead>
<tr>
<th>EPC</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
<th>C8</th>
</tr>
</thead>
<tbody>
<tr>
<td>e123</td>
<td>mouse</td>
<td>NULL</td>
<td>111105 T0815</td>
<td>NULL</td>
<td>IS.b: 120.22.1.17</td>
<td>NULL</td>
<td>12</td>
<td>NULL</td>
</tr>
<tr>
<td>e243</td>
<td>pen</td>
<td>NULL</td>
<td>111205 T1200</td>
<td>IS.e: 40.41.51.127</td>
<td>NULL</td>
<td>15</td>
<td>NULL</td>
<td>SC2</td>
</tr>
</tbody>
</table>
5.2.2. EPCglobal Security Analysis

Sarma et al. (2002) examined RFID security and identified several general RFID threats including tracking, spoofing, session hijacking, replay attacks, man-in-the-middle attacks, and power interruption. RFID security is a popular topic for research. In average, there are more than 50 articles published on conferences and journals every year since 2005; however, few researches have been done on EPCglobal security.

The EPC-enabled system’s vulnerabilities refer to several aspects. We developed a comprehensive taxonomy to illustrate the vulnerabilities of the EPCglobal systems, shows in Table 5.2. The vulnerabilities are divided into two major categories, the non-technical factors and the technical factors.

<table>
<thead>
<tr>
<th>Non-technical factors</th>
<th>Technical factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Lack of security &amp; privacy awareness</td>
<td>- Low cost requirement</td>
</tr>
<tr>
<td>- Lack of security &amp; privacy education</td>
<td>- Computation capability</td>
</tr>
<tr>
<td>- Lack of technology education</td>
<td>- Wireless interface</td>
</tr>
<tr>
<td>- Absence of related laws</td>
<td>- Tag size</td>
</tr>
<tr>
<td>- Incomplete standards</td>
<td>- Read range</td>
</tr>
<tr>
<td>- Human</td>
<td>- Unprotected software</td>
</tr>
<tr>
<td>- Unclear Business rules</td>
<td>- EPCglobal infrastructure</td>
</tr>
</tbody>
</table>

Non-technical factors mainly relate to human, social and business strategy aspects. First, many people do not have much security and privacy awareness, since they hardly detect that they are in dangers or they are attacked by enemies and they usually don’t care their privacy neither. In most cases, they do not even clearly know what privacy is. Lacking of security and privacy awareness is more dangerous than an intended attacker. The reason that people lack of security and privacy awareness are most likely that they lack of security and privacy education, so that
people cannot protect themselves. Moreover, RFID is a new technology; thus, most people have no idea how it works and thus bringing in the security and privacy troubles.

Besides the awareness and education parts, the absence of related laws/regulations is another critical vulnerability. Currently, it's unclear how restrictive any potential regulations would be. The absence of laws may encourage the attacker’s behaviors. Third, the development of RFID and EPCglobal standards are still in progress. The incomplete standards cannot avoid security and privacy problems from the very beginning design and deployment.

Furthermore, information systems are always vulnerable to some uncertain human factors, because it is hard to predict people’s behavior. There are plenty of human relevant works within the EPCglobal system, comparing with computer and machines, people will be more possible to make mistakes consciously or unconsciously. In addition, the hostile employees from the intra-organization may cause huge damages.

Another notable vulnerability is the unclear business rules. Currently, we hardly find any well-defined business rule for the EPC-enabled system. EPCglobal standard defines the underlying information system infrastructure that is composed of various hardware and software systems. However, an explicit business rule that coordinating the daily business activities of the whole supply chain with the EPC-enabled system is still missing at this moment. Business rules aim at directing every key EPC-enabled system related business process within the organization as well as among different supply chain partners. For example, business rules will solve some critical issues such as data collection, information sharing, data redundancy and disposal issues, and also handle the exception events. Data collection rule indicates locations and time interval (where and when) of the data collection. Normally, the data only needs to be collected when it is necessary, such as the product’s status has changed, e.g. when the product leaves the warehouse or the temperature exceeds normal range. When the RFID tag works along with sensors, the data
collection rule will also regulate what data should be collected, e.g. temperature, pressure, humidity, etc. Information sharing policy is relevant to business security. It allows the supply chain partners to manage their information in a safe but also efficient way, and thus protecting their organizational privacy. The policy should point out who can access what information. For instance, a retailer may maintain record of multiple suppliers who are not allowed to see each other’s information, so that preventing the suppliers to form price alliance. Data redundancy is another severe issue due to RFID will generate huge amount data. The business rule will regulate the system to dispose the data which is no longer needed automatically in order to minimize the data volume. While a complete business rule is finalized, it is possible to achieve the integration of existed business processes into the EPCglobal infrastructure.

From the technical point of view, there are seven vulnerabilities: low cost requirement, wireless interface, weak computational capability, small tag size, the various read ranges, unprotected software system, and EPCglobal infrastructure.

Because of the commercial value of EPCglobal applications and its extremely larger scale of deployment, the corporations have to keep the cost as low as possible in order to gain more profits. Thus, the EPCglobal system (especially the EPC tag) was kept at the simplest design, from the manufacture materials selection to circuit design. As results, it is not able to resist many hardware attacks, e.g. physics destructions, electronic destructions by electronic pulse, tag cloning, etc. In addition, due to the same reason, EPC tag has very limited computational capability. It hardly performs sophisticated cryptographies and other high computational demand implementations, such as complex authentication protocols. It is, therefore, vulnerable to many threats, such as eavesdropping. Wireless transmission interface between the reader and tag is another notable vulnerability. Attackers can easily perform many attacks that aiming at this nearly
unprotected wireless interface, while the tag cannot execute advanced programs to protect the communications.

Besides, the small tag size is another vulnerability of the system. Because the tag is too small to notice, sometimes the tag holder doesn’t even know they have been tagged or the tag has been removed from the item. It will vulnerable to some attacks, such as clandestine tracking, tag remove / replace. Moreover, many problems are caused by various read ranges of the reader. The read range is one of the most important factors in RFID security due to the wireless transmission. Juels (2006) identified four types of read ranges that the distances are various from centimeters to kilometers. Since the reader’s read range can be extremely large in some cases, people will not know the tags have been scanned by adversaries. Additionally, some software are not well protected that could be attacked by the sophisticated adversaries. For instance, the hacker may use malware or hack the database or the back-end system, and etc.

Currently, the underdeveloped EPCglobal infrastructure is one of the most significant technical related vulnerabilities for EPCglobal system. The EPCglobal architecture framework document states that because of the level of acceptable risk differs widely from application to application, there is no standard security solution that can apply to all systems, and thus the EPCglobal architecture framework cannot be pronounced secure or insecure (EPCglobal 2010a). It has been recognized that EPCglobal standards do not concern the security and privacy issues much, so that the security mechanisms is absent from the infrastructure. Attackers can compromise the EPCglobal system easily through internet.
5.2.3. EPCglobal System Threats

We located the whole system into three levels. According to the ascending order, from the low to the high level are RFID sub-system level, network level, and business/strategy level respectively. We identified four entities in the RFID sub-system level: EPC tag, reader, back-end system, and interface/communication channel. For the network level, there are four entities, Middleware, EPC-IS, EPCONS, and interface/communication channel. We summarized all the threats, as well as mapping them each levels and entities, illustrates in Table 5.3.

From the table we can see there are six threats target on the business/strategy level, competitive espionage, social engineering, privacy threats and targeted security threats are indicated in (Mitrokotsa et. al., 2009), plus the information sharing threats that we discussed above, and the insider attacks.

The business competitors can easily obtain confidential information due to the lack of security mechanism in the EPCglobal system. And attackers may simply use some social engineering method to perform attacks, e.g. taking advantage of human nature such as kindness, fearness, trust, desire to help, and obey to authority. In addition, attackers can track and profile individuals based on the information they steal from the RFID tags. Thus, the adversaries are able to locate the victims, or get the information about the victims’ habits and preference, or other kinds of private information. Moreover, the adversaries can perform attack when certain target appears, e.g. RFID-bomb that is active when people are holding passports from certain countries. Information sharing threats occur when unauthorized access to the confidential information, as we analyzed above. The insider attack from disgruntled employees is very hard to prevent, and may cause huge loses.
As we mentioned in the former sections, EPCglobal network level consists of four fundamental elements: EPC middleware, EPCIS, EPCONS, and the EPCDS. The security issues

<table>
<thead>
<tr>
<th>BUSINESS/STRATEGY LEVEL</th>
<th>INTERFACE/CHANNEL</th>
<th>NETWORK LEVEL</th>
<th>BACK-END SYSTEM</th>
<th>RFID SUB-SYSTEM LEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>EPCONS</td>
<td>Reader</td>
<td>EPC tag</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Man-in-the-middle</td>
<td>- Unauthorized tag reading</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Eavesdropping</td>
<td>- Tracking</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Spoofing</td>
<td>- Tag cloning</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Replay attack</td>
<td>- Tag tamper</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Cache Poisoning</td>
<td>- Tag misuse/replace</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DoS</td>
<td>- Tag remove/destroy</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Betrayal by trusted server</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- IP Address Spoofing</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Packet Interception</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>EPCIS</td>
<td>Back-end breakdown</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Unauthorized access</td>
<td>- Man-in-the-middle</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Data Injection</td>
<td>- Eavesdropping</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DoS</td>
<td>- Spoofing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Malware</td>
<td>- Replay attack</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Middleware</td>
<td>- Correlated Keys</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Interface/Channel</td>
<td>- Unauthorized access</td>
<td>- Side-channel attack</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Data Injection</td>
<td>- RF data modification</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DoS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Malware</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**As we mentioned in the former sections, EPCglobal network level consists of four fundamental elements: EPC middleware, EPCIS, EPCONS, and the EPCDS. The security issues**
of EPCglobal network are very similar as the traditional issues from the internet and cyber security. Middleware is essentially an application server, while EPC-IS consists of EPCIS repository which can be considered as database server, and EPC-IS accessing application which can be considered as application server (Konidala et. al., 2007). Thus, the threats toward Middleware and EPC-IS are similar as the threats aim at traditional servers. So that, unauthorized access, data injection, DoS, and Malware are the most well known threats (Bertino et. al., 2004). Since EPCONS was built using the same technology as the DNS, the security threats related to DNS server are also applicable to ONS, such as cache poisoning, DoS, betrayal by trusted server, IP address spoofing, packet interception (Atkins and Austein, 2004). The interface security refers to the internet communication channel security, where eavesdropping, replay attack, spoofing and man-in-the-middle attacks are the most notorious attacks.

There have been plenty of paper discussed the threats in RFID sub-system level, we summarized and combined the similar threats, and mapped them into each entities. Among those, tracking, eavesdropping, and denial of service/jamming are the most obvious threats to RFID sub-system as well. On the other hand, threats like tag misuse, elevation of privilege, and correlated keys have earned least attentions. But it does not mean those threats are not important, for example, we believe the tag misuse is a very special threat specific to the RFID sub-system. The RFID tag is generally working as the identification of person/item. Thus it is very important to establish a certain link between the tag and the tag owner to secure the ownership and prevent the tag misuse threat. However, neither low-cost EPC tag nor people/products can efficiently perform cryptographic operations, even the very simple ones. Resultantly, the tag misuse or identification theft can occur in a walk. Thus, researchers should pay more attentions to those less popular threats in the future also. Among those three levels, the RFID sub-system level threats are
specific to the RFID technology, while the threats toward other two levels are common to other computer or information systems.

5.2.4. Possible Countermeasures

The countermeasures can be deployed in all the three levels of EPCglobal system, the RFID sub-system, the EPCglobal network and the business/strategy level.

In the sub-system level, the solutions can be divided into two categories, non-protocol and protocol approaches. Most studies are focused on the protocols, especially the on-tag protocols. Table 5.4 summarizes the potential sub-system threats countermeasures. The on-tag protocols are the most important research areas nowadays.

<table>
<thead>
<tr>
<th>Categories</th>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-protocol</td>
<td></td>
</tr>
<tr>
<td>Physical disable</td>
<td>Kill</td>
</tr>
<tr>
<td></td>
<td>Blocker tag</td>
</tr>
<tr>
<td>Logical disable</td>
<td>User control</td>
</tr>
<tr>
<td></td>
<td>Sleeping</td>
</tr>
<tr>
<td>Protocol</td>
<td></td>
</tr>
<tr>
<td>Security Agent</td>
<td>Watch dog</td>
</tr>
<tr>
<td></td>
<td>Guardian</td>
</tr>
<tr>
<td>On-tag Protocol</td>
<td>RFID Enhancer Proxy</td>
</tr>
<tr>
<td></td>
<td>Hash lock</td>
</tr>
<tr>
<td></td>
<td>Hash chain</td>
</tr>
<tr>
<td></td>
<td>Scalable protocols</td>
</tr>
<tr>
<td></td>
<td>Non-cryptographic</td>
</tr>
</tbody>
</table>

The non-protocol approach mainly refers to disable the tag completely or partially. However, these approaches have certain limitations that they eliminate the entire after sale benefits. Security agent, a personal device that works between the tag and reader, can specify a number of policies that readers must comply with. Early versions for such mediating systems, called “watchdog” device, have been suggested in (Floerkemeier et. al., 2004). Rieback et. al.
(2005) proposed the RFID Guardian which can enforce various policies when interacting with readers. The weakness of this approach is that the Guardian must always be alert in protecting tag responses from unauthorized read attempts. It has to either allow reader queries or actively block tag answers which may not always be feasible. RFID Enhancer Proxy, which assumes the identities of tags and simulates them in the presence of reading devices by continuously relabeling the identifiers transmitted by tags, is an improvement over the RFID Guardian (Juels, 2005). The disadvantages are that the tag identities need to be partially generated by the tag and match portions of its true ID. The common shortcoming of security agent is the performance highly relies on the security policy set.

Protection between tag and readers is mainly implemented by on-tag security protocols. The two fundamental protocols are hash-lock scheme (Weis et. al., 2003) and hash chain scheme (Ohkubo et. al., 2003). But they share a same shortcoming that they are not suitable for large-scale environment. The scalability problem was addressed successfully in (Molnar et. al. 2005). It used a correlated key method to manage the huge amount of keys. However, this improvement does not enhance the security features of the RFID sub-system, but the efficiency. Instead, it compromised the RFID security and privacy due to it may be suffered by the correlated key threat. Vajda and Buttyan (2003) proposed a set of challenge response authentication protocols without using cryptographic operations that can be supported by low cost RFID tags. However, Defend et al. (2007) demonstrated that these protocols can be easily broken, and moreover they cannot prevent tracking. Worth noting both the non-protocol and the protocol approaches cannot completely solve the security problems. The low cost requirement dramatically limits the security performance. And most current protocols do not suit for the large scale usage. Thus, the study of upper layer solutions would be necessary.
In the network level, the interface/communication channel can be secured by using existing security protocols, such as the Internet Protocol Security (IPSEC), Transport Layer Security (TLS), and Secure Sockets Layer (SSL). The traditional countermeasures for the middleware and EPC-IS include implementing system authentication, authorization, and access control mechanisms. Malware can be mitigated by installing anti-virus software and firewall or intrusion detection system, and performing security audit, and backup important data. Code review, input validation and input/output encoding can effectively reduce the impacts of packet injection. Additionally, the X.509 architecture can be used for reference to prevent unauthorized access threats. EPCONS can be protected by similar security mechanisms that used for DNS, such as VPN, DNS security extensions (DNSSEC), and TLS.

At this moment, the business/strategy level threats can be hardly solved by technical solutions only. Thus, as we have discussed before, in order to prevent the threats in this level, we should consider the high level solutions, e.g. perfecting the EPCglobal standards, enforcing the relevant legislations, and educating people, and so on.

5.3. Proposed Access Control Model for EPC Discovery Service

Inside an EPCglobal network, objects move among network partners, the associated EPC event data are captured by different information resources and stored locally. EPCDS is sitting on the top of data repositories to manage the data retrieving, distributing and security functions. It simplifies the data exchange process by offering a service that links all the relevant information of a certain object across the supply network, and therefore allows the participants to proactively manage their supply chains and ultimately gain further benefits that EPCglobal promised.
Usually, the EPCDS is deployed in the middle of information resources and system users, where is also the right place to integrate the security enforcement. Figure 5.2 depicts the framework of the proposed integrated discovery service and access control model. The Service Requestor represents the client, who invokes the Lookup Service to request the EPC event data. The Lookup Service Module uses the EPC key and the Item Table to locate the target EPCIS, and then forwards the request. The EPCIS’s Relationship-Based Access Control Module captures the request and starts interfering. Based on the relationship type (RT) and the pre-defined security policy set, it generates the authorization decisions and returns the decisions to the Federated Access Control Module where the authorization decisions are evaluated and aggregated. The federated decisions are generated and sent to each EPCIS, who uses the Query Interface to extract the target EPC event data from the EPCIS repository. The EPCDS will collect all the sub-results that returned by every EPCIS, and present the final result to the client.

![Figure 5.2 An Integrated DS and Access Control Framework](image)

In the EPCDS design, we assume that all the EPC keys are distributed into the EPCDS nodes evenly. Each node maintains an Item Table, which needs to be explained beforehand. Item Table records the movement path of each product. Table 5.5 shows an example of Item Table, in which the prime key is “EPC”, and another attribute “Pi” indicates the locations of this product.
This location indicator $P(i)$ is used to store the IP addresses of every EPCIS who ever possesses information about this product, where “$i$” is a positive integer that indicates the relative position of each organization (EPCIS) in the supply chain. The value format of this attribute is $<\text{IS.name:IP}>$. If an EPCIS repository belongs to company A, it will be denoted by IS.a, therefore $<\text{IS.a:47.92.101.23}>$ indicates that the IP address of company A’s EPCIS repository is 47.92.101.23. Because an object cannot physically present at more than one location, when each company publishes this EPC key into the EPCDS Item Table serially, the Item Table can reflect the correct movement path for each product, such as P1 must be the upstream company of P2, while P3 must be the downstream company of P2. This innovative Item Table design provides a solid evidence to the system to identify the relationship between entities. In this case, each Item Table row represents one supply chain for a specific product.

Table 5.5 An Example of Item Table

<table>
<thead>
<tr>
<th>EPC</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
</tr>
</thead>
<tbody>
<tr>
<td>e123</td>
<td>IS.a:47.92.101.23</td>
<td>IS.b:120.22.1.17</td>
<td>IS.c:23.62.18.28</td>
<td>IS.d:37.22.101.23</td>
<td>IS.e:12.207.13.17</td>
<td>IS.f:223.6.138.28</td>
</tr>
<tr>
<td>e243</td>
<td>IS.d:58.2.211.156</td>
<td>IS.e:40.41.51.127</td>
<td>IS.a:47.92.101.23</td>
<td>IS.g:58.2.211.156</td>
<td>IS.h:40.41.51.127</td>
<td>IS.i:47.92.101.23</td>
</tr>
</tbody>
</table>

EPCDS network contains three types of entities: individual client or consumer, organizational client, and EPCDS. The individual client/consumer is an independent system user that has no partnership with any business entity/organization. They request services from the system, but do not offer any service. The organizational clients can be referred to either service provider or service user, and usually their roles can be switched between these two. In order to control the communications between different types of entities, EPCDS is expected to provide certain security enforcement, e.g. access control mechanism that contains authentication and authorization to protect both of the service users and providers.
5.3.1. Federated Authentication and Key Publishing Security Control

A proper authentication mechanism is the prerequisite for the authorization mechanism. In order to offer the authentication for securing the EPCglobal network and the communications among clients, EPCIS and EPCDS, the X.590 certificate can be utilized to specify and prove the identification of each entity. X.590 is based on the Public Key Infrastructure (PKI), which contains three core components: Certificate Authority (CA), Certificate Revocation Lists (CRLs), and Registration Authority (RA) (Housley et al., 2002). Its major functions include issuing certificates, revoking certificates, key management, digital signatures, encryption and decryption etc. These functions make sure the electronic communications are not intercepted, read and altered by unauthorized entities during transmission. Verifying the identity of the parties involved in an electronic transmission ensures that these parities cannot be denied for their involvement in the transaction. And finally provides integrated, authenticated, confidential and non-reputable communications (BRIDGE, 2008).

The outcome of this federated authentication is to figure out the business relationship between the subject and object entities. This relationship is the foundation for the object entity to make the authorization decision. In order to achieve this, the entities must be able to prove their identities, and thus a certification architecture is employed to assign certificates. The design of Entity Certificate Architecture and the concept of Entity Relationship Types, as well as the Federated Authentication Workflow, deserve detailed discussion.

5.3.1.1. Entity Certificate Architecture

Our proposed PKI-based CA architecture (see Figure 5.3) consists two root CAs: the DS root CA and the EPCIS root CA, denotes by CA.DS and CA.IS respectively. They are responsible for identifying and issuing X.590 certificates for each DS and IS entity. Individual consumer will not be able to obtain any certificate, since it is not possible to issue certification for them, and
there is no meaning to identify this type of system users. Generally, information resource will grant the lowest level of access privilege to them. If a user initiates a service request to EPCDS network without a certificate, it will be considered and treated as an individual consumer user.

![PKI Model based CA Architecture](image)

**Figure 5.3 PKI Model based CA Architecture**

The DS root CA and IS root CA will first sign each other, and then identify and issue certificates for their affiliated entities. Organizational entity will issue certificates to their employees through their EPCIS. In this way, each organizational user will obtain a certificate with the unique identifier (UID), e.g. IS.a.director.user1. In order to prove identity and establish trust, entities will need to exchange their certificates to establish mutual authentication.

### 5.3.1.2. Entity Relationship Types

It is obvious that the business relationship between two entities is extremely important for access control. Under the EPCglobal context, there are four types of business relationship, which enable organizations to set up the corresponding security rule:

- RT0 refers to the situation that the subject is an individual user or consumer to the target object.
- RT1 refers to the situation that the subject affiliates to an entity who does not have any business relationship with the organizational entity that the target object affiliates to.
• RT2 refers to the situation that the subject affiliates to an entity who has certain business relationship with the organizational entity that the target object affiliates to. This implies that the subject and object are in the same supply network.

• RT3 refers to the situation that the subject affiliates to the same organizational entity as the object does.

The DS system is responsible for identifying the entity relationship types between two business parties, according to the subject’s certificate and the EPCDS’s Item Table. If a subject fails to surrender a legitimate certificate, the system will consider the subject as an individual user or consumer. No matter which object it intends to access to, the relationship type will be RT0 between these two entities. This type of relationship is very easy to be identified and handled, because all the target objects will give this type of subjects the lowest level of permission. If a subject successfully presents a legitimate certificate, the system will consider the subject as an organizational level user. Based on the certificate, the DS is able to know which organization the subject affiliates to. By searching the Item Table, DS can then identify whether this subject is within the object’s supply network or not. If the subject comes from an organization that outside the supply network, it will be labeled as RT1 type, otherwise it will be labeled as RT2 type. If the subject logs into the EPCIS server or virtual private network (VPN), and requests information within the intranet, it will be considered as RT3, which is handled by each organization internally. This type of security concerns will not be discussed in the dissertation.

Please note that the major players of the EPCglobal network are the organization users, rather than the individual users/consumers who under RT0 relationship. The consumers will most likely to get some basic EPC event information from EPCIS, however the main purpose of EPCglobal network is to facilitate information sharing for the current or potential business partners, and thus it mainly servers entities that are under RT1 and RT2 relationships.
Depends on the business needs of each entity, both RT1 and RT2 can be further divided into more sub-relationship types. For example, refer to the EPCDS Item Table in Table 5.5. This Item Table indicates that there are nine companies, Companies A to I, and they all associate with two EPC codes, e123 and e243. According to this table, we can infer the business relationships for each entity pair. It is obvious that, for EPC e123, Companies A, B, C, D, E and F are under relationship RT2, because they locate in the same row of e123. It means, for this specific item e123, these companies are in the same supply chain and have certain business relationship with each other. Moreover, this RT2 can be further divided to whole-stream, upstream, or downstream relationships, according to the position indicator “Pi”. The upstream and downstream relationships can be further divided into immediate and non-immediate upstream/downstream relationships.

Whole-stream relationship for a certain EPC means they all maintain EPC event data for certain EPC. This relationship can be recognized, if any two companies are in the same row. In this example, companies A, B, C, D, E and F are under whole-stream relationship for item e123 and companies D, E, A, G, H and I are under whole-stream relationship for item e243.

Upstream relationship for a certain EPC means the subject publishes EPC event data before the object does. Formally, it can be defined based on P(i). If X locates at the P(i)th column in the Item Table, while Y locates at the P(i+x)th column of the same row, where \(\{i, x\} \in \mathbb{Z}^+\) (\(\mathbb{Z}^+\) denotes the set of all positive integers), then X is the upstream company for Y. For instance, according to this Item Table when Companies A, B, C, D and E intend to access information about e123, they all have the upstream relationship with Company F.

This upstream relationship can be further divided to immediate upstream and non-immediate upstream relationships. The immediate upstream relationship means the two entities do business directly and the subject is one position ahead the object, while non-immediate
upstream relationship means the two entities are upstream relationship but do not do business directly, i.e. the subject is more than one positions ahead the object. If \( x=1 \), then \( X \) is the immediate upstream company of \( Y \), whereas if \( x>1 \), then \( X \) is the non-immediate upstream company for \( Y \). In this example, when company \( A \) tries to access information about \( e123 \), it will get the immediate upstream relationship with company \( B \), and non-immediate upstream relationship with companies \( C, D, E \) and \( F \).

Downstream relationship means for a certain EPC the subject publishes EPC event data after the object does. It also can be defined based on \( P(i) \). If \( X \) locates at the \( P(i) \)th column in the Item Table, while \( Y \) locates in the \( P(i-x) \)th column of the same row, where \( \{ i, x, (i-x) \} \in \mathbb{Z}^+ \), then the \( X \) is the downstream company for \( Y \). For instance, when company \( F \) intends to access \( e123 \), it is under the downstream relationship with companies \( A, B, C, D \) and \( E \).

Similarly, the immediate downstream means the two entities do business directly and the subject is one position after the object, while non-immediate downstream relationship means the two entities are downstream relationship but do not do business directly, which means the subject is more than one positions after the object. According to Table 4.1, when company \( F \) tries to access \( e123 \), it will be the immediate downstream company for \( E \), and the non-immediate downstream companies for \( A, B, C \) and \( D \).

RT1 relationship refers to two entities that do not have any business relationship for a certain EPC. This can be identified by searching the EPCDS Item Table, if \( X \) published a specific EPC in the Item Table, while \( Y \) does not, then \( X \) and \( Y \) are under RT1 for this EPC. For example, when companies \( B \) and \( C \) try to access information about \( e243 \), they will be under the RT1 with companies \( A, D, E, G, H \) and \( I \), because \( B \) and \( C \) are not present in the row of \( e243 \). Moreover, RT1 can also be divided into complete RT1 and incomplete RT1. Complete RT1 means the two entities do not maintain EPC event data for any EPC code together, which can be identified by
going through the entire Item Table. If the subject and object are not present in any row together, then they are complete RT1. While incomplete RT1 means that although the two entities do not maintain EPC event data for the target EPC, they do maintain other EPC together. For example, when companies B and C try to access information about e243, they are under complete RT1 relationship with companies G, H and I, and incomplete RT1 relationship with companies A, D and E, because A, B, C, D and E possess EPC event data for e123 together.

It is important to point out that each company has the entitlement to decide and define their own security rules. It means that they can choose which types of business relationships will be involved. Some company may consider all basic business relationship types, e.g. RT0, RT1, and RT2, while others may want to set up more sophisticated security regulation, and therefore, choose to include more sub-relationship types. This complicated and flexible scheme allows each independent organization to customize its own security mechanism.

5.3.1.3. Federated Authentication Workflow

Figure 5.4 illustrates the EPCDS federated authentication process. When the organizational users log into their host EPCIS servers to initiate service requests to the EPCDS network (S1), they will obtain the certificate of the EPCIS they belong to (S2), and then prove their identity to the EPCDS by rendering their identification documents, i.e. the certificates, which contain the attributes that associate with this user, e.g. serial number, issuer, subject, subject Public Key information, roles or positions, and certificate signature. When DS server receives the authentication request, it will response to the requester by its own certificate (S3). The first mutual authentication is complete by now. Once the EPCIS confirms the identity of the requested DS node, it can then send out the original request (S4). The requested DS node will then find out who is the destination DS node, and ask for mutual authentication by passing its certificate. While confirming the certificate, the destination DS node will response by its
certificate to complete the second mutual authentication (S5). The user’s certificate and the original request can now be forwarded to the destination DS (S6). Thereafter, the destination DS node will find out who is the target information resource, and initiates query by exchanging their certificates for the third mutual authentication (S7).

Once the certificate was verified, the Destination DS node forwards the request to the information resources, which are protected by the federation partnership (S8). If the user is required to be authenticated, the information resources will act as the Service Providers (SP), and generate a SAML2 Authentication Request, <AuthnRequest> to the Identity Provider (IDP), i.e. the Destination DS node in this case (S9). According to user’s certificate and the EPCDS Item Table, the IDP is able to authenticate the user, and in the meanwhile identify the relationship between this user and the information resources (S10), and thereafter to generate a SAML2...
Authentication Response, <AuthnResponse>, for each different information resources (S11). The <AuthnResponse> carries the RT type attribute, which is considered as the most important authentication information for federation, and other attributes can also be included, e.g. subject’s role attribute, environmental attributes, and time attributes etc. Based on the <AuthnResponse> and the pre-defined security rules, the authorization model is able to make the access decision (S12). At the end, the object EPCIS will return the result to the user via the same path (S13). In this process, the subject’s identification and sensitive data should not be disclosed to the information resource directly.

```xml
<saml:AuthnRequest
 xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" Version="2.0"
 ID="a09ec8cc-305-4566-acfd-fb943e2aa66"
 IssueInstant="$(java.util.GregorianCalendar(GregorianCalendar.getInstance()).get Instance())"></saml:Issuer
 Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
 http://www.EPCISx.com/
 </saml:Issuer>
 <saml:Subject>
  <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
   Manager.User1
  </saml:NameID>
  <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:sender-vouches"></saml:Subject>
 </saml:Subject>
 <saml:AuthnStatement>
  <saml:AuthnContext>
   <saml:AuthnContextClassRef>
    urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified
   </saml:AuthnContextClassRef>
  </saml:AuthnContext>
 <saml:AuthnStatement>
  <saml:RequestedAuthnContext xmlns:saml="urn:oasis:names:tc:SAML:2.0:protocol" Comparision="exact">
   urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
  </saml:RequestedAuthnContext>
 </saml:AuthnRequest>

Figure 5.5 Sample SAML <AuthnRequest>
```

This method increases the flexibility of the network and protects the privacy of the two parties. In this process, a user is required to provide identification documents to EPCDS only, and
thus the original certificates and the detailed identity information of this user will not be known by the service provider. In the meanwhile, the destination DS node is responsible for collecting the results that subject to the security permission. Figure 5.5 and Figure 5.6 show the sample SAML <AuthnRequest> and <AuthnResponse> messages. In this example, we can see that the SAML <AuthnRequest> message is issued by the information provider EPCISx, i.e. <saml:Issuer>http://www.EPCISx.com</saml:Issuer>, while the SAML <AuthnResponse> is issued by the destination DS node DestinationDS, i.e. <saml:Issuer>www.DestinationDS.com</saml:Issuer>. The main purpose of this SAML transaction is to deliver the most important information RT, which is carried by the <Saml:Attribute RTType="RT"> element of the SAML response, as we can see that the value of this element is 2, which mean the subject and object entities are under RT2 relationship.

Figure 5.6 Sample SAML <AuthnResponse>
5.3.1.4. Key Publishing Security Control Mechanism

This mutual anonymous authentication procedure is considered as an advanced security feature that protects the privacy for both requester and target EPCIS. However, when the security control of the key publishing is missing, it also brings a security vulnerability that has been discussed in R13. When a malicious entity publish a forged key into the Item Table, it is able to insert itself into this supply chain and establish a fake business partnership with the target EPCIS. In this scenario, the target EPCIS is not able to aware this threat due to the anonymous authentication. As a conclusion, a security enforcement for the key publishing is necessary.

In order to achieve this, it needs to employ the EPC Object Name Service (ONS), which is a well-developed component of EPCglobal that provides a simple lookup service that constantly points to the product manufacturer, who is placed at the head position (P1) of the Item Table as well as the supply chain. In addition, it also requires the EPCIS repository to reserve an attribute that points to its downstream company.

When the Publishing Service is invoked, the key publishing security control mechanism starts to interfere. Firstly, it will search the Item Table to identify whether this is the first time registration. If it is, the publisher must be the manufacturer of this product. The Security Module has to verify the identity of this publisher from the EPCONS. Once confirmed, the Publishing Service can be proceeded, otherwise it will be terminated. In another case, if it is not the first time registration, it still needs to verify the publisher’s identity with the reserved attribute of the EPCIS repository that currently locate at the tail position of the Item Table. Once confirmed, the Security Module can approved the key publishing operation, otherwise it should deny the operation. This verification will be conducted only once, when an entity attempts to publish a key.

As can be seen that the mutual anonymous authentication offers a way to protect the privacy of the EPCDS participants, while it also causes the security concern that the malicious
entity can disguise itself as a legitimate supply chain partner. This issue can be solved by the introduced identity verification mechanism that effects at the key publishing process, and therefore restricting the illegal publisher to publish EPC keys into the EPCDS Item Table.

5.3.2. Relationship-based Federated Authorization Model

Once object EPCIS receives the <AuthnResponse>, it is able to decide the RT type of the user, and then starts the authorization. The process of authorization composes two stages: the relationship-based authorization and federated authorization. The relationship-based authorization is conducted by the object entity to generate the authorization decision, which is enforced at its own EPCIS repository. This stage allows each organization to control its own security, and enables the database level access control. The relationship-based authorization can satisfy the security demands, if target data only locates in one object EPCIS. However, there are usually multiple hosts for one EPC, and therefore the interdependent security issue will become a concern. In order to solve this issue, federated authorization needs to be conducted by the EPCDS. The second stage is expected to evaluate all the decisions that made by each EPCIS and generate a final decision.

5.3.2.1. Relationship-based Authorization

As explained, in order to make the authorization decision, the object EPCIS needs to know the exact RT type of the subject entity, and then based on all of the security policies that imposed on this RT type to make the authorization decision. The RT type is depicted by the subject and object entity ontologies. Let us use the ontologies in Figure 5.7 as an example to illustrate how entity ontologies influence the security policies and authorization.
Both subject and object ontologies are defined and maintained by the object entity for the authorization purpose. The subject ontology classifies the subjects based on their entity relationship types RT. For the host object entity, node S0 represents the subject who is under RT0 relationship, node S1 refers to the subject who is under RT1 relationship, node S2 indicates the subject who is under RT2 relationship, and node S3 shows the subject who is under RT3 relationship. The object ontology classifies the EPC event data based on the attributes.

The security policies are imposed on these two ontology trees. Each node will be controlled by one or more security policies, and the child node will inherit the security policy from its parent, and enforce its own security policy. The security policy of a node usually clarifies which source can or cannot be accessed. For example, if one policy decides that S1 (RT1) is restricted to access node O2 (Price), this policy will be propagated to all the child nodes of S1 and O2. Thus, S4 cannot access O2, and since O4 and O5 inherit security policy from O2, and as a result, S1 and S4 are restricted to access O4 (Buying Price) and O5 (Selling Price). Besides the RT types that defined in subject and object ontologies, the authorization decisions are also based on other attributes, such as the environmental attributes, which usually infer the semantic

Figure 5.7 Subjects and Object Ontology Models
meanings, e.g. current date or time. It is important to known each object’s security needs and which types of environmental attributes will be involved in the decision making.

In this dissertation, the authorization policy is represented by a three tuple expression (sub, obj, a). Subject (sub) and object (obj) refer to either a concept or an instance in the ontology. They also indicate the access decision when a client wants to access an EPC event data resource. In the policy, “a” means the action (e.g. view, modify, delete and etc.) that subject will execute on the object. For simplicity, we assume that there is only one possible action, access, i.e. a=access. The value of “a” belongs to {1, 0}; if a==1 means the access is permitted, otherwise, the access is denied.

The authorization policy is generated based on the security rules. The security rules are defined by security administrator in each individual organization and are written in the human readable format. For example, the security administrator of company A sets up a security rule that individual clients cannot access any data in A’s EPCIS repository. Based on this security rule, the corresponding authorization policy is (S0, O1, 0). According to the policy inherit feature, all the child nodes of O1 will inherit this policy. The complete policy set that derived from this rule will be {(S0, O1, 0), (S0, O2, 0), (S0, O3, 0), (S0, O4, 0), (S0, O5, 0)}.

The three parameters indicates that authorization policy is written in machine understandable format so that they can be used by the system for authorization directly. Although the security rules can only be understood by human, instead of machines, they can be translated into Semantic Web Rule Language (SWRL) (Horrocks et al., 2004) by the security administrator. The main advantage of using SWRL is to express rules in terms of Web Ontology Language (OWL) and provides support for complex relationships between properties, i.e. reasoning the entity relationship (Shen, 2009). The aforementioned security rule example can be interpreted into SWRL as: inRT(0, ?sub) => denyAccess (?sub, ?obj).
The SWRL rule axiom is composed by an antecedent, such as inRT(0, ?sub), and a consequent, such as denyAccess(?sub, ?obj). When a SWRL rule is activated, the system is able to generate the corresponding authorization policy automatically.

Next a more concrete example is presented to explain the authorization policy and decision generation. Assume the subject and object relationship are based on the ontologies that shown in Figure 8. The security administrator of company A makes the following security rules for its EPCIS repository:

- Security Rule #1: Individual consumers cannot access any EPC event data.
- Security Rule #2: Users under RT1 cannot access Price data.
- Security Rule #3: Users under RT2 cannot access Buying Price data.
- Security Rule #4: Users under RT2 with attribute Engineer cannot access Price data.
- Security Rule #5: Internal users can access all the EPC event data.

After defining the security rules, they will be translated into SWRL format as:

- SWRL Rule #1: inRT(0, ?sub) => denyAccess (?sub, ?O)
- SWRL Rule #2: inRT(1, ?sub) => denyAccess (?sub, ?O2)
- SWRL Rule #3: inRT(2, ?sub) => denyAccess (?sub, ?O4)
- SWRL Rule #4: inRT(2, ?sub) ∧ attributeRole (?sub, Engineer) => denyAccess (?sub, ?O2)
- SWRL Rule #5: inRT(3, ?sub) => permitAccess (?sub, ?O)

Thereafter, based on the SWRL rules and policy inherit principle, the corresponding authorization policies matrix will be generated as:

- (S0, O1, 0), (S0, O2, 0), (S0, O3, 0), (S0, O4, 0), (S0, O5, 0)
- (S1, O1, 1), (S1, O2, 0), (S1, O3, 0), (S1, O4, 0), (S1, O5, 1)
- (S4, O1, 1), (S4, O2, 0), (S4, O3, 0), (S4, O4, 0), (S4, O5, 1)
By now, all the authorization policies have been set up, and the authorization decisions have been made according to those policies. In case, there are two or more conflict security policies that enforced on one pair of subject and object, the decision making should follow the “Deny Takes Precedence” or “Permit Takes Precedence” principal to resolve the conflict. Each organization needs to define the confliction resolution principle for each of its object entity node. Let us use the “Deny Takes Precedence” principal as an example for illustration: If there are two different security policies are imposed on (S5, O2) and (S5, O5) pairs, according to the confliction resolution principle, the authorization decision for these two pairs will be (S5, O2, 0) and (S5, O5, 0).

5.3.2.2. Federated Authorization Model

Although each object entity can make the authorization decision based on its own security rules, it is not sufficient to ensure the system secure due to the aforementioned Interdependent Security problem. The relationship-based access control model provides a secured way to access single independent and autonomous information resources, as each EPCIS defines their own security rules and policies; however it ignores the security of other entities. While these entities are within the same supply network and share the relational data, one organization’s authorization decision may compromise the security of other organizations, because of the interdependent security relationship. Hence the current authorization model is not enough to protect the sensitive information. In order to eliminate the Interdependent Security threat, a
A federated authorization mechanism is needed. The proposed federated authorization model handles the security from the holistic angle and takes care of all the relevant entities.

The basic idea of the federated authorization model is to implement an extra security layer in front of the end user, where the best place will be at the network level – EPCDS. The goal of this security component is to find a way to evaluate the results/policies/decisions that returned by each independent EPCIS, and then aggregate them to a correct final result. There are more than one possible ways to achieve this, for instance, the Result Aggregation method (RA), the Security Rule Aggregation (SRA) method and Decision Aggregation (DA) method.

- **Result Aggregation (RA):** RA is the most intuitionistic method to achieve the goal. It is executed after the relationship-based authorization module makes the decision and also completes the result extraction. At this time, all of the sub-results will be returned to the EPCDS by each individual resource. The sub-results need to be evaluated and aggregated into a final result by the federated authorization enforcement that is deployed at the EPCDS side, and then displayed to the end user. Figure 5.8 shows the working process of this method.

![Figure 5.8 The Result Aggregation Process](image)
Step 0: Each EPCIS receives the authorization request from the EPCDS network.

Step 1: The EPCIS’s policy enforcement point (PEP) module captures the request, and asks for an access control decision from policy decision point (PDP).

Step 2: PDP queries the relevant rules and policies from the Policy Repository.

Step 3: Policy Repository returns the corresponding policies.

Step 4: PDP makes the authorization decision based on the policies, and returns this decision to the PEP module.

Step 5: PEP queries the result from the EPCIS Repository, according to the access control decision.

Step 6: EPCIS Repository returns the corresponding result.

Step 7: The sub-results are sent back to the Result Aggregation Module of the EPCDS for the final result aggregation.

- Security Rule Aggregation (SRA): SRA achieves the goal in a different way, it is executed before relationship-based authorization module makes any decisions. This approach needs to collect all the relevant security policies that are defined by different information resources. In order to eliminate the conflicts, these independent policies are reviewed, reasoned and aggregated at the EPCDS side into one unified policy that is applicable to the entire data federation. This federated security policy is used to generate the authorization decisions, in order to regulate the behavior of the query engines. Figure 5.9 shows the working process of this method.
Figure 5.9 The Security Rule Aggregation Process

Step 0: Each EPCIS receives the authorization request from the EPCDS network.

Step 1: The EPCIS’s policy enforcement point (PEP) module captures the request, and sends a command to the Policy Repository.

Step 2: According to the received command and request, each Policy Repository sends the relevant rules to the Security Rule Aggregation Module for reasoning and aggregation.

Step 3: The PDP makes and returns a holistic authorization decision based on the aggregated rules and the generated policies.

Step 4: PEP queries the result from the EPCIS Repository, according to the authorization decision.

Step 5: EPCIS Repository returns the corresponding result.

Step 6: PEP sends the correct result to the EPCDS.

- Decision Aggregation (DA): DA is executed after the relationship-based authorization module makes the decision, but before each sub-results are extracted. DA is required to
collect all the access control decisions, which are generated by each individual information resource. The decisions are generated based on the pre-defined security rules, which are independent to each other. The federated authorization module is responsible for reasoning and aggregating these decisions into a final federated access control decision, which will be used to extract data from each independent data repository. Figure 5.10 shows the working process of this method.

Figure 5.10 The Decision Aggregation Process

Step 0: Each EPCIS receives the access request from the EPCDS network.

Step 1: The EPCIS’s policy enforcement point (PEP) module captures the request, and asks for an authorization decision from policy decision point (PDP).

Step 2: According to the received command and request, each Policy Repository sends the relevant rules and policies to the PDP for reasoning.

Step 3: The PDP makes an authorization decision based on the aggregated policies.

Step 4: PDP returns the decision to the EPCDS’s Decision Aggregation Module.
Step 5: The Decision Aggregation Module collects, reasons and aggregates all the sub-decisions at the EPCDS side, and then pushes the final decision back to each PEP.

Step 6: PEP queries the result from the EPCIS Repository, according to the authorization decision.

Step 7: EPCIS Repository returns the corresponding result.

Step 8: PEP sends the correct result to the EPCDS.

The Result Aggregation model requires the EPCDS to pull out all the sub-results, which may contain the sensitive data, and thus cause the security conflicts, i.e. the interdependent security problem. However, it simply aggregates those sub-results and display the final result to the user and has no clue to filter these sub-results nor to resolve the confliction. As such, the RA approach cannot fulfill the security requirement that handling the interdependent security problem.

Comparing with the RA approach, the SRA approach is much more advanced due to the addition of policy aggregation. To implement that, the EPCDS is finally able to resolve the conflicts and eliminate the interdependent security problem. However, in order to aggregate the security policies, it needs to extract all the relevant policies that defined by various organizations into EPCDS itself for evaluation. In addition, in order to reasoning these rules, their context environment and semantic meaning need to be investigated and taken into consideration. In this case, the EPCDS needs to prepare a considerable volume of extra space to maintain these policies and their corresponding hosts. Moreover, it will also require plenty of extra processing time for EPCDS to aggregate these policies and make the decisions. This extra time that EPCDS spends for policy evaluation will be much more than each individual EPCIS spends, because it converts a decentralized operation into a centralized one, which will definitely cost more
resources. Although we disregard all these disadvantages to make such procedure possible, the security policies are still considered as business sensitive data, which should not be disclosed.

Apparently, the Decision Aggregation model the (DA) is more feasible, secure and efficient than the other twos. It is imposed after the EPCIS generates the authorization decisions, and before the sub-results are pulled out. EPCDS is responsible for the decision aggregation, rather than the decision generation. After the reasoning, the EPCDS is able to generate a final decision that satisfies the security demands of all the relevant entities. In this case, it ensures that neither sensitive data nor the security policies will be exposed to and reviewed by other parties, while the decisions can only reveal very limited information about the business. Therefore, the aggregation procedure we propose is based on the decision aggregation rules (AgR), which examine the ontology of the data sources and the semantic meaning of each sub-decision.

5.3.2.3. Decision Aggregation Model

The Decision Aggregation Model is derived from the object source data ontology. The source data is the target information that desired by the user. In the authorization policy, the source data points to the target objects. The class level of the source data ontology also represents the class level of the object ontology, which shows in Figure 5.11.

We classify the source data into two major categories, the Static and Dynamic Data. The Static Data represents the data that stays constant over the entire supply life cycle, regardless of the varying of time, ownership or other conditions. Although these data are maintained by multiple organizations, they remain the same. For example, the name/brand and the description of the object belong to static type of data, since they are identical in every entity. Static Data is redundant due to its nature. Redundancy occurs when all the parties return the duplicate Static Data as part of their sub-results. Obviously, there is no meaning to include every piece of them into the final result. The Decision Aggregation rule needs to be able to remove these redundancies.
The dynamic data, on the other hand, keeps changing due to different times, ownerships or other conditions. These data are most likely different for each independent entity. For instance, in the matter of location, different companies will probably store their product in different warehouses and storage racks, especially by using different coding schema for location representation. One portion of the dynamic data relates to each other in the matter of ownerships, while the other portion does not. The former ones are defined as relational dynamic data, and the later ones are called non-relational dynamic data. The typical relational dynamic data includes the price data and time data. It is easy to understand that the selling price of the immediate upstream company is the same as the buying price of the immediate downstream company. Similarly, it is not hard to figure out the approximate inbound time, if anyone knows the outbound time of its immediate upstream company. The reason is that there is a strong deducible connection between the selling and buying price as well as the outbound and inbound time inside the supply network. Even though the two companies are not under the immediate upstream or downstream relationship, it is still able to infer a range of the price or time, because the values of these attributes in the upstream companies will be always, in the normal cases, smaller than those in the
downstream companies, and vice versa. Thus, it will bring security concerns, if any of the supply partner does not want to disclose this type of data, while others do.

According to this class level source data ontology, every instance level object data in the EPCIS repository will fall into a concept level class, which will become an instance of this concept. These instances will become effect on the decision aggregation process.

The Aggregation Rule (AgR) is the fundamental principle of the Decision Aggregation. It is responsible for eliminating the result redundancy and solving the interdependent security threat. The AgR is divided into three major principals to address three different conditions:

- Aggregation Rules I (AgR I): If the target object falls into one of the Static Data classes, a final authorization decision will be generated according to the returning decisions from each individual entity. The decision was generated subjects to the “Deny Takes Precedence” principle. The client will receive only one unique final result, instead of a union of multiple sub-results, in order to eliminate the result redundancy. Two conditions need to be considered under this principal:

  1) When all the returning decisions are permitted, a uniform final decision “permit” should be generated, i.e. in the final authorization decision \((s, o, a)\), \(a=1\). Thereafter the sub-results will be collected from each corresponding data repository, and then pruned into the final result, which will contain only one copy of the sub-result.

  2) When there is any party returns the “deny” as its decision, the final decision “deny” will be generated due to the “Deny Takes Precedence” principle, i.e. in the final authorization decision \((s, o, a)\), \(a=0\). Therefore, no sub-results will be collected from any data repository, and the final result will be “Denied”.

For example, when a client requests the brand information of a product, which means the target object falls into the \(<\text{Brand}>\) concept level class. According to the Source Data ontology,
<Name/Brand> is one of the Static Data types. Assume that there are two companies, “C1” and “C2”, are maintaining data about this product, and the two instance level data for this product will be <C1::Brand:Thinkpad> and <C2::Manufactory:LenoveThinkpad> that stored in C1 and C2, respectively. If both of them issue the authorization decisions as “permit”, the aggregated final authorization decision will be “permit” as well. The sub-results will be extracted and pruned into one copy of final result <Brand:LenoveThinkpad>. If any of them issues the authorization decision as “deny”, the aggregated final authorization decision will be “deny”. Accordingly, none of the data will be extracted and returned from the data repositories.

- Aggregation Rules II (AgR II): If the target data falls into one of the Non-Relational Dynamic Data classes, there will be no change to the returning decisions, and they will be used as the final decisions directly. Each returning decision is independent and will not affect other decisions. EPCDS will not generate a uniform final decision, instead of forwarding each returning decisions back to the original entity. The sub-result collection complies with the final decisions and all returning sub-results will be aggregated and presented to the client. Two conditions need to be considered under this principal:
  1) When the returning decisions is “permit”, the EPCDS will forward this decision back as the final decision. According to this final decision, the sub-results will be collected from each corresponding data repository, and then aggregated into the final result union, which will contain every piece of the sub-result.
  2) When there is a party returns the “deny” as its decision, the EPCDS will forward this decision back to the original entity as the final one. Therefore, no sub-result will be collected from this data repository, while other returning decision will not be influenced by this deny decision. Sub-results with “permit” decision can be retrieved from the data repositories.
For example, when a client requests the location information of a product, which means the target object falls into the <Location> concept level class. According to the source data ontology, <Location> is one of the Non-Relational Dynamic Data types. Assume that there are two companies, “C1” and “C2”, maintain data about this product. The instance level data will be <C1::Location:SC1> and <C2::Loc:Zone3> that stored in C1 and C2 respectively. If both entities make the “permit” decision, EPCDS will issue “permit” as their final authorization decisions. The sub-results will be extracted and aggregated as a union, and thereafter make the final result {<C1::Location:SC1>, <C2::Loc:Zone3>} available for the client. If any of these two company issues “deny” as the authorization decision, only the permitted data will be extracted. For instance, C1 returns the “deny” decision, while C2 makes the “permit” decision, the final decisions will be “deny” for C1 and “permit” for C2. The client will only receive <C2::Loc:Zone3> as the final result.

- Aggregation Rules III (AgR III): If the target data falls into one of the Relational Dynamic Data classes, the returning authorization decisions need to be evaluated based on the “Deny Takes Precedence” principle. If one entity returns “deny” as its decision, EPCDS will need to look up the returning decision of its relational data, and alter this decision into “deny”. The Item Table will be the evidence for locating the immediate upstream and downstream companies. The sub-results will be extracted from each EPCIS repository based on the corresponding final decision, and then present to the client. Two conditions need to be considered under this principal:

1) When all the returning decisions are “permit”, the EPCDS will forward these decisions back as the final decision. Thereafter the sub-results will be collected from each corresponding data repository, and then aggregated into the final result union, which will contain every piece of the sub-result.
2) When there is any party returns the “deny” as its decision, the EPCDS will forward this decision back to the original entity as its final decision, and also evaluate its corresponding relational data by finding its immediate upstream or downstream entities. When the relational data is located, the final authorization decision for this data will be altered to “deny” due to the “Deny Takes Precedence” principle. Therefore, both of the relational data pair will not be collected from the data repositories, while other sub-results can be retrieved from the data repositories, if their final decisions are “permit”.

For example, when a client requests the price information of a product, which means the target object falls into the <price> concept level class. According to the source data ontology, <price> is one of the Relational Dynamic Data types. Assume that C1, C2 and C3 are three companies in one supply network, while C1 is the immediate upstream company for C2, and C2 is the immediate upstream company for C3. The instance level data will be \{<C1::bp:10>, <C1::sp:20>, \} \{<C2::bp:20>, <C2::sp:30>\} and \{<C3::bp:30>, <C3::sp:40>\} that stored in C1, C2 and C3, respectively. If all the parties issue “permit” as their authorization decisions, there will be no change for the final authorization decisions. The sub-results will be extracted and aggregated as a union, and thereafter make the final result \{<C1::bp:10>, <C1::sp:20>, <C2::bp:20>, <C2::sp:30>, <C3::bp:30>, <C3::sp:40>\} available for the client. If any of the organization refuses to disclose part of this type of data, the access to its relational data should be also restricted, e.g. when C2 issues “deny” as the authorization decision for its buying price, the final decision for C2’s buying price as well as C1’s selling price will be “deny” due to the fact they are relational data pair. The final result has to comply with this final decision, and thus it will be \{<C1::bp:10>, <C2::sp:30>, <C3::bp:30>, <C3::sp:40>\}.
Please note that the third rule is only applicable to relational data, which is defined as the related data pairs that stored in the immediate upstream and downstream entities, e.g. the selling price from the immediate upstream company and the buying price form the immediate downstream company are relational data for each other. To locate the immediate upstream and downstream company pairs, EPCDS will need to look up its Item Table. The Item Table of the above example is illustrated in Table 5.6. Every two adjacent columns of the Item Table indicate an immediate upstream and downstream company pair.

Table 5.6 Illustration of the Immediate Upstream and Downstream Relationship

<table>
<thead>
<tr>
<th>EPC</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
</tr>
</thead>
<tbody>
<tr>
<td>e123</td>
<td>IS.c1: 47.92.101.23</td>
<td>IS.c2: 120.22.1.17</td>
<td>IS.c3: 23.62.18.28</td>
</tr>
</tbody>
</table>

Once the company pair is recognized, the relational data will be identified. According to the source data ontology, each company can present its object data as a tree. If a relational data object locates on a left subtree of a parent node, then the EPCDS will look for its immediate upstream company and locates the corresponding relational data at the right subtree of the same parent node, which shows in Figure 5.12 (a). After finding out the relational data pair, the “Deny Takes Precedence” principal will be executed, which will alter the authorization decision that effects the other subtree, which shows in Figure 5.12 (b). If the a relational data object locates on a right subtree of a parent node, the EPCDS will look for its immediate downstream company and locates the corresponding relational data at the left subtree of the same parent node, and then performs the same operation to this node.
Whenever the EPCDS receives a “deny” authorization decision for a relational data type from the policy repository, it needs to do the aforementioned procedures. When all the “deny” authorization decisions are evaluated, the final authorization decisions can be issued, and thereafter the sub-results can be pulled out and aggregated to the final result.

5.3.2.4. Real Case Illustration

A real case example is used to demonstrate the effectiveness of the Relationship-based Federated authorization model. This example compares the EPCDS results in three different cases: no security control is applied, only the Relationship-based Authorization Module is applied, and both Relationship-based and Federated Authorization Modules are applied. Let us assume that product e123 of the supply chain associated with two companies A and B, who are under immediate upstream and downstream relationships to each other. Figure 5.13 shows the simplified sketch. Their IS repositories, EPCIS.a and EPCIS.b, maintain EPC event data about (C1) Name, (C2) Buying Price, (C3) Selling Price, and (C4) Location.

Case 1: No Security Control. When there is no security control applied at all, the EPCDS will bypass the Relationship-Based and Federated Authorization Modules, and simply extract and combine the sub-results from the EPCIS repositories. The EPCDS input and output are shown in Table 5.7, where we can see that all the EPC event data are presented to the client.
Table 5.7 EPCDS Input and Output of Case 1

<table>
<thead>
<tr>
<th>EPCDS Input</th>
<th>e123</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPCDS Output</td>
<td>{Name: mouse, Buying Price: 10, Selling Price: 12, Location: SC4} ∪ {Name: mouse, Buying Price: 12, Selling Price: 15, Location: SE0}</td>
</tr>
</tbody>
</table>

Figure 5.13 A Real Case Example

Case 2: Only the Relationship-Based Authorization Module (RBAM) is Applied. In this case, the RBAM needs to obtain client’s RT information from the EPCDS, and then based on each organization’s security rules and policies to make the authorization decisions. These decisions will be used to extract the EPC event data from the repositories. The returning sub-results will be simply combined and presented to the client.

Assume the client CLT is under RT2 relationship with the information resources. Company A and B, sets independent security rules and policies for the RT2 type of client as follows:

- **Company A (EPCIS.a):**
  - Security Rule #1: RT2 client can access product name. That is, Policy #1: (RT2, C1, 1)
  - Security Rule #2: RT2 client can access buying price. That is, Policy #2: (RT2, C2, 1)
  - Security Rule #3: RT2 client can access selling price. That is, Policy #3: (RT2, C3, 1)
  - Security Rule #4: RT2 client can access location. That is, Policy #4: (RT2, C4, 1)

- **Company B (EPCIS.b):**
o Security Rule #1: RT2 client cannot access product name. That is, Policy #1: (RT2, C1, 0)
o Security Rule #2: RT2 client cannot access buying price. That is, Policy #2: (RT2, C2, 0)
o Security Rule #3: RT2 client cannot access selling price. That is, Policy #3: (RT2, C3, 0)
o Security Rule #4: RT2 client cannot access location. That is, Policy #4: (RT2, C4, 0)

The input and output of each module are shown in Table 5.8, when the RBAM is applied. As we can see that, although the security policies of company B consider (C1) Name and (C2) Buying Price as confidential data, they are still disclosed to the client by company A, because the Federated Authorization Module is not applied in this case.

Table 5.8 The Input and Output of Each Module of Case 2

<table>
<thead>
<tr>
<th>EPCDS Input</th>
<th>e123</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBAM Input</td>
<td></td>
</tr>
<tr>
<td>IS.a</td>
<td>&lt;e123, RT2&gt;</td>
</tr>
<tr>
<td>IS.b</td>
<td>&lt;e123, RT2&gt;</td>
</tr>
<tr>
<td>RBAM Output</td>
<td></td>
</tr>
<tr>
<td>IS.a</td>
<td>[{RT2, C1, 0}, {RT2, C2, 1}, {RT2, C3, 1}, {RT2, C4, 1}]</td>
</tr>
<tr>
<td>IS.b</td>
<td>[{RT2, C1, 0}, {RT2, C2, 0}, {RT2, C3, 0}, {RT2, C4, 0}]</td>
</tr>
<tr>
<td>EPCDS Output</td>
<td>{Name: mouse, Buying Price: 10, Selling Price: 12, Location: SC4} ∪ { null}</td>
</tr>
</tbody>
</table>

Case 3: Both RBAM and Federated Authorization Module (FAM) are Applied. The RBAM outputs will be evaluated by the FAM, based on AgR and source data ontology, where C1 falls into the static data class, C2 and C3 fall into the relational data class, and C4 falls into the Non-Relational Data class. The federated authorization decisions will be generated and returned to each EPCIS. The sub-results will be extracted based on these decision, and then aggregated and presented to the client. Table 5.9 shows the input and output of each module. As we can see that, two of the IS.a’s authorization decisions have been alerted by FAM. The final result has shown that, the security demands of both companies A and B have been satisfied.
Table 5.9 The Input and Output of Each Module of Case 3

<table>
<thead>
<tr>
<th>EPCDS Input</th>
<th>e123</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBAM Input</td>
<td>IS.a</td>
</tr>
<tr>
<td></td>
<td>&lt;e123, RT2&gt;</td>
</tr>
<tr>
<td>RBAM Input</td>
<td>IS.b</td>
</tr>
<tr>
<td></td>
<td>&lt;e123, RT2&gt;</td>
</tr>
<tr>
<td>RBAM Output</td>
<td>IS.a &lt;(RT2, C1, 1), (RT2, C2, 1), (RT2, C3, 1), (RT2, C4, 1)&gt;</td>
</tr>
<tr>
<td></td>
<td>IS.b &lt;(RT2, C1, 0), (RT2, C2, 0), (RT2, C3, 0), (RT2, C4, 0)&gt;</td>
</tr>
<tr>
<td>FAM Input</td>
<td>IS.a</td>
</tr>
<tr>
<td></td>
<td>((RT2, C1, 1), (RT2, C2, 1), (RT2, C3, 1), (RT2, C4, 1))</td>
</tr>
<tr>
<td>FAM Input</td>
<td>IS.b</td>
</tr>
<tr>
<td></td>
<td>((RT2, C1, 0), (RT2, C2, 0), (RT2, C3, 0), (RT2, C4, 0))</td>
</tr>
<tr>
<td>FAM Output</td>
<td>IS.a</td>
</tr>
<tr>
<td></td>
<td>((RT2, C1, 0), (RT2, C2, 1), (RT2, C3, 0), (RT2, C4, 1))</td>
</tr>
<tr>
<td></td>
<td>IS.b</td>
</tr>
<tr>
<td></td>
<td>((RT2, C1, 0), (RT2, C2, 0), (RT2, C3, 0), (RT2, C4, 0))</td>
</tr>
<tr>
<td>EPCDS Output</td>
<td></td>
</tr>
<tr>
<td></td>
<td>{Buying Price: 10, Location: SC4} ∪ {null}</td>
</tr>
</tbody>
</table>

Table 5.10 compares the final results of all three cases that output by the EPCDS. We can clearly see that the RBAM and FAM effect on the final result.

<table>
<thead>
<tr>
<th>Final Results</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Name: mouse, Buying Price: 10, Selling Price: 12, Location: SC4} ∪ { Name: mouse, Buying Price: 12, Selling Price: 15, Location: SE0}</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>{Name: mouse, Buying Price: 10, Selling Price: 12, Location: SC4} ∪ {null}</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>{Buying Price: 10, Location: SC4} ∪ {null}</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.4. Analysis and Performance Evaluation

In this section, we implement and analyze the proposed relationship-based federated access control model in term of its security, performance and scalability.

5.4.1. Security Analyses

We analyze if the proposed model really satisfies the seven security requirements presented in section 3.1. Firstly, we identified four major types of entity relationships (RT types), and some of them were further divided into sub-relationships. According to our proposal, the
model is able to prove the business relationship between the subject and object entities, the object entity then makes the authorization decision for the subject entity based on each RT type. The EPCDS employs an authentication model and the Item Table to automatically validate the RT type. Thus, R11 is fully satisfied by our design. Secondly, under the federation of EPCDS, both the subject and object entities render their own certificates to the EPCDS network, instead of exchanging the certificates with each other. In this way, EPCDS only works as an intermediate broker to provide mutually authentication, without disclosing the actual identities to the other parties. The only information that the object entity will know about the subject is the RT type, which is encapsulated in the SAML response <AuthnResponse> message. Similarly, the object can only view the pruned final result through EPCDS’s interface. Thus, the double anonymous authentication (R12) is achieved. The consequent key publishing security issue (R13) is solved by a publishing control mechanism, which guarantees the authenticity of the published data. Thirdly, the relationship-based authorization module is based on the security rules that defined by each object entity. These security rules are only responsible for the security of its own EPCIS repository. The database level security (R14) is managed by its host organization in this design. After the object entity makes authorization decision, an additional security layer deployed at EPCDS side is applied to support network level security (R15). The fifth requirement (R16) is fulfilled by the relationship-based authorization itself, which imposes the authorization decision on each instance level object. An instance level object represents one cell in the EPCIS database, which enables the fine-grained access control. Finally, the interdependent security threat (R17) is prevented by the proposed federated authorization mechanism. The decision aggregation approach is used to evaluate all the returning decisions against the Aggregation Rules (AgR), which provides detailed instructions on how to resolve the confliction among the returning
decisions. In short, the proposed relationship-based federated access control model can support all six security requirements for EPCDS.

5.4.2. Performance Evaluation

To assess the relative performance of our proposed relationship-based federated access control model, we develop a prototype that based on the integrated discovery service and access control framework, as shown in Figure 5.2, to test the performance of Decision Aggregation Model. Here, the Service Requestor mimics the client to generate an initiation message \(<\text{EPC key}, \text{RT}>\) to invoke the Lookup Service. The first parameter is the target EPC key, and the second parameter is the RT between the subject and object. The Lookup Service Module uses the EPC key to locate the target EPCIS, and then forwards the request. The EPCIS’s Relationship-Based Authorization Module captures the request and starts interfering. Based on the RT type and the pre-defined security policy set, it generates the authorization decisions and returns the decisions to the Federated Authorization Module, which is deployed at the EPCDS side. These decisions are evaluated and aggregated based on the AgR. The federated decisions are generated and sent to the Query Interface of the EPCIS. The interface extracts the EPC event data from the EPCIS repository and returns them to the Lookup Service Module, who is responsible for presenting the final result.

Our experiments take into consideration of two variables, the amount of the requests (Rn) that input to the prototype and the number of EPCIS nodes (ISn) that deployed in the network. We conducted 16 sets of tests, where 10, 100, 1000 and 10000 requests (Rn) combining with 10, 50, 100 and 200 EPCIS nodes (ISn), respectively. For each experiment of request, the subject is a randomly selected user instance that may have different RT types with the target subject, and the
target object is a randomly picked EPC key that already registered in the EPCDS. The evaluations are conducted by measuring the time efficiency of the prototype. In the test, we keep one variable constant and change the size of the other to measure how these two variables impact the time of loading the returning authorization decisions (TdL) and the time of generating the final decision (TdG). In order to be statistically convincible, we run 20 rounds test cases for each different variable combinations, and calculate the average time elapse. The average time accurate to millisecond. The test bed is deployed in a desktop server with an Intel Quad Core I7-2600 CPU (3.40 GHz) and 16 GB RAM, running in Windows 7 64x Professional operation systems. The prototype is codded in Java using the Java Virtual Machine version jre1.8.0_25. The EPCIS Repository and Policy Repository are constructed in SQL databases. And the Policy Decision Point (PDP) runs on the Tomcat server.

Table 5.11 summarizes the computational results. As shown, when Rn or ISn are increased, both TdL and TdG will increase accordingly. This is not surprise. However, we notice that the time of loading the returning decisions is much longer than the time of generating the final decision. This can be explained as when loading the decisions, the security model needs to make the connection with external application and databases. In order to retrieve the returning decisions, the PDP needs to query to the policy repository located in an external SQL database, which will consume much more time. On the contrary, the decision generation is fairly efficient as the Item Table and source data ontology are pre-defined and pre-loaded into the logic of the DA module. Here, Item Table and source data ontology are used for the purpose of locating the immediate upstream and downstream companies as well as the Relational Data pairs.
Figure 5.14 (a) and (b) depict the variation of the average time of decision loading (TdL) with different number of requests and nodes. As shown, the computational times increase when the number of requests and nodes increase. This is because the time complexity under both conditions are approximately in linear O(n). Also, when the number of node (ISn) increases, the TdL shows less increasing rate. This indicates that ISn has less impacts on the TdL as compared to the Rn. The reason is clear because the security module has to process the request one by one, and each of them consumes similar time for loading the returning decisions and the TdL is the sum of all individual time consumptions. However, when the amount of node grows, the TdL will not increase significantly. Because the target objects are only diffused in limited number of nodes, thus, not all the new added nodes are involved into the procedure.
Figure 5.14 Average Time Based on Rn and ISn Variables

Figure 5.14 (c) and (d) show the variation of the average time of decision generating (TdG) with different number of requests and nodes. Similar to TdL situation, the time complexity under both conditions are approximately O(n), thus, the curves display almost in linear relationship. Also, Rn shows stronger impacts on TdG; that is, when there are more number of requests, the time takes to generate the decision will increase more rapidly. The reason of this phenomenon is also similar to the previous scenario.

Figure 5.15 (a) and (b) depict the variation of the total time consumed for processing, which is the sum of both decision loading (TdL) and generating (TdG). From the curves, we can observe that the differences in the increasing rate between these two test cases become smaller. This is because decision loading usually takes much more time than the decision generating. When we sum up these two factors, the one with bigger base (i.e. TdL) will show more impacts into the final result.
Figure 5.15 Average Total Time Trend Based on Rn and ISn Variables

Figure 5.16 (a) and (b) compares both TdL and TdG under different conditions. For both test cases, the decision generating takes less time than the decision loading, and the curves based on the ISn variable show lower gradients in the figures. Another factor influencing this result is the size of the Rn and ISn. In the experiment, we increase the request size ten times for each test, while the node size has much smaller changes. And the Rn is also much larger than the ISn. The reason why using different orders of magnitude for the two variables is because we are trying to simulate the real world situation. In real world the EPCglobal network is an open distributed network that is available to all the users that include not only the supply network partners but also the huge amount of individual consumers. The number of organizations that linked by one particular target object could be limited, and therefore, the Rn will probably be larger than the ISn in a few orders of magnitude.
5.4.3. System Scalability

It is well-known that the EPCglobal network is a large scale system and thus the security solution must fit into this situation as well. We analyze the scalability from the adopted EPCDS infrastructure’s point of view, and also evaluate the scalability against the size of requests and EPCIS nodes.

Firstly, The EPCDS is an open distributed system that deploys on multiple DS nodes globally, which can handle the large scale requests. Our security mechanism will be implemented on each of these node. Statistically each EPCDS node will receive approximately equal amount of requests to balance the load and decentralize the user request, this benefit will apply to each security component as well. Assume that there are totally N EPCDS nodes that deployed worldwide, each of them will receive (Rn/N) requests, which will be process by the security component. As a result, the proposed access control model is also decentralized into the EPCDS network infrastructure, and thus it is able to fit into the large scale application. Secondly, from the results of our experiment, we can clearly see that when the number of request (Rn) increases, the total processing time will also increase nearly linearly. However, the increases are in an
acceptable range (about 53.7 to 79.7 ms). Thirdly, the results also show that when the size of the EPCIS node (ISn) increase, the total processing time will show smaller increases. The increase rate is also much lower than the Rn’s, which demonstrates having reasonable scalability. In short, the scalability of the proposed relationship-based federated access control model is acceptable when the sizes of request and node grow.

5.5. Conclusion

In this chapter we identify and summarize the main security concerns and requirements that faced by the EPCDS networks. Especially, we identify and explore the key publishing security issues and the interdependent security threat in details, and propose a novel relationship-based federated access control model as a solution. We provide evidence to show that this model can fulfill all seven security requirements that we have revealed, and that it can be easily integrated with the EPC Discovery Services to overcome these security issues and eliminate redundant results. This new security countermeasure has been proved as a flexible solution that can be applied under the dynamic and diversity environment. It can also deliver the maximum protection to all the network participants efficiently and effectively.

Unique features of the proposed solution include (1) It supports mutual anonymous authentication, which protects the privacy of the network users and the services providers. (2) The consequence key publishing security issues is pointed out, and solve the a publishing control mecanism. (3) This model provides a way to automatically identify the entity relationship between the subject and the object, which is the most critical evidence for decision making. (4) Each independent information resource can define its own security rules, and make the independent authorization decisions according to these rules and the entity relationships. In this
way, the object entity is able to control its own EPCIS repository, which stores valuable business data. And (5) The authorization decisions that made by each information resource are only responsible for protecting its own data that may compromise its partner’s security. The federated access control mechanism aims to solve this confliction by employing the Aggregation Rules based Decision Aggregation model to evaluate all the independent decisions. These features highlight the main contribution of this study to this emerging field.

Although the performance of the proposed relationship-based federated access control model is acceptable. The scalability of this model can be further improved. One of the feasible approaches is to implement “cache” for the decision processing. Another strategy is to implement a relational data index such that the decision loading and generating time can be monitored and reduced accordingly.
Chapter 6: Conclusion and Future Work

6.1. Conclusion

In this dissertation, we have explored the research issues and challenges in designing a sophisticated discovery service for global supply chains.

We first explore the EPCDS solutions in EPCglobal environment. According to current studies, we propose a comprehensive list of EPCDS design requirements, which cover the topics from three aspects: the functionality and service, network performance, and security and privacy design requirements. We present a novel model to address first two challenges. The model employs a peer-to-peer infrastructure with a consistent hashing enabled key assignment function to construct a highly scalable, efficient, and effective EPCDS network, which enables the network to distribute a huge amount of EPC keys into a small number of DS nodes, and therefore allows the users to easily publish and retrieve the target information. A fully deployed EPCDS network can satisfy the global scale usage expectation. We also adopt and enhance the result aggregation method to ensure the query accuracy. The two-layer decentralization design of the Item Table and DS table further spread the EPC event data into EPCIS repositories that avoids the possible failure issues, and a backup plan is introduced to enable the failure recovery mechanism, which further enhances the system availability and reliability. Besides the basic service, e.g., lookup and publishing services, the proposed Advanced Service Module can also support the advanced services, e.g., subscribing and pushing services. The performance of the model is evaluated by the implemented prototype. Four metrics—load balance, publishing and lookup time, and path length—have been tested under a diverse set of conditions. Compared with other existing EPCDS designs, this newly proposed model can offer more sophisticated services, enhance the network performance, and secure the operation procedures. The experimental results show that the system
demonstrates good load balancing ability where the largest volume of EPC keys is able to be distributed into a number of DS nodes evenly by the key assignment function. The publishing and lookup time decrease/remain stable, when there are more companies that join the network or a larger key space is applied. The network path length is shown to be a constant value $O(2)$, no matter how the company prefix number or the EPC key length increase.

A relationship-based federated access control model is proposed to integrate with the proposed EPCDS network infrastructure. The mutually anonymous authentication protects the privacy of the network users and the services providers, but in the meantime, it brings the key publishing security threat, which requires a key publishing control mechanism to verify the authenticity of the publisher. Since the authorization decision relies highly on the relationship between the service requesters and providers, this study classifies these relationships and offers a way to identify the entity relationships. Under this highly connected network environment, the security decision of one entity may compromise the security of another entity. To solve this condition, the federated access control mechanism is introduced. The performance of the security model is evaluated by the prototype. To the best of our knowledge, this is the first work to support the mutually anonymous authentication mechanism, and the first work to address the key publishing security threat and the interdependent security issues.

The experimental results illustrate that the model is able to load the generated decisions efficiently and effectively. The computational times increase when the number of requests and nodes increase in linear $O(n)$. We also note that the number of nodes has less impact on the decision loading as compared to the number of requests, because each request consumes a similar time for loading the returning decisions, and the loading time is the sum of all individual time consumptions. However, when the amount of nodes grows, the loading time will not increase significantly since not all the new added nodes are involved in the procedure. The time
complexity of decision generating is also approximately $O(n)$, where the number of requests shows stronger impacts on decision generation. Additionally, the experimental result shows that decision generating takes less time than the decision loading, and we can clearly see that when the number of requests increases, the total processing time will also increase linearly in an acceptable range, and on the contrary, when the size of the EPCIS node increases, the increase rate of the total processing time is much lower. More importantly, the security module is able to integrate with the EPCDS infrastucture seamlessly, and thus it can inherit the good scalability.

6.2. Future Work

This dissertation work opens some research issues on large-scale EPC event data management, and autonomous threat detection for EPCglobal network. We leave these as future research topics. Specifically, we can extend this work to the following directions:

i) Optimizing the proposed EPCDS infrastructure to further improve network performance. Although the performance of the proposed model demonstrates good scalability, it needs to be further improved, especially when the security module is integrated and the network performance may be restricted. A feasible approach is to implement “cache” for the decision processing. It can be implemented at each EPCIS, which will enhance the decision returning capability. Another strategy is to implement a relational data index and put it into the cache, such that the decision loading and generating time can be monitored and reduced accordingly.

ii) Developing an autonomous threat detection plan for the EPCglobal network. The proposed security model is based on entity relationship, and once the relationship is determined, the authorization decision will be made. However, in case that service
requestor is hacked or the communication channel is compromised, the system is not able to respond appropriately. Hence, an intrusion detection mechanism is needed. One possible solution is to apply the Continuous Hidden Markov Model (CHMM) to recognize service requestors’ behaviors. The CHMM is able to predict the abnormal query probe, based on the pattern inferred from the previous queries, and thus to eliminate the threats at an earlier stage.
References


BRIDGE. Bridge project high level design for discovery services. 2007a. Available online at: bridgeproject.eu/data/File/BRIDGE%20WP02%20High%20level%20design%20Discovery%20Services.pdf


EPCglobal. EPC Information Services (EPCIS) version 1.0.1 specification. 2007b. Available online at: http://www.gs1.org/gsmp/kc/epcglobal/epcis/epcis_1_0_1-standard-20070921.pdf

EPCglobal. EPCglobal Object Name Service (ONS), v.1.0.1. 2008. Available online at: www.gs1.org/gsmp/kc/epcglobal/ons/ons_1_0_1-standard-20080529.pdf

EPCglobal. EPCglobal architecture framework, v1.4. 2010a. Available online at: www.gs1.org/gsmp/kc/epcglobal/architecture/architecture_1_4-framework-20101215.pdf

EPCglobal. EPC tag data standard, v.1.5. 2010b. Available online at: www.gs1.org/gsmp/kc/epcglobal/tds/tds_1_5-standard-20100818.pdf


EDUCATION:

- B.E in Engineering (Software Engineering), Beijing Institute of Technology, China. (2002 - 2006).

RELEVANT PUBLICATIONS:


OTHER PUBLICATIONS: