DATA PROTECTION OF RFID-BASED DISTRIBUTED STORAGE

RAKESH SAINI

A THESIS

IN

THE CONCORDIA INSTITUTE FOR INFORMATION SYSTEMS ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF APPLIED SCIENCE IN INFORMATION SYSTEMS SECURITY

CONCORDIA UNIVERSITY

MONTRÉAL, QUÉBEC, CANADA

AUGUST 2009

© RAKESH SAINI, 2009

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-67238-9
Our file Notre référence
ISBN: 978-0-494-67238-9

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

ABSTRACT

Data Protection of RFID-based Distributed Storage

Rakesh Saini

Radio Frequency Identification (RFID) has been emerged as one of the most promising technologies used as an automatic data collection and information storage technology in vast number of applications. One of the biggest hindrances in the wide adoption of this technology is the challenge in security. There have been extensive studies on RFID security, in particular authentication and privacy issues. In most protocols, the discussions focus on scenarios that RFID tags are used mainly for tracing or identification, and the access to data stored on RFID is enforced through authentication. Recently, there is a rise in interests of using RFID tags as distributed storage, e.g., storing floor plans which can be used by fire fighters during emergencies. In this new type of applications, quite often, XML (eXtensible Markup Language) is employed since it has been considered as a de-facto standard to store and exchange information on the Internet and through other means. This research proposes to securely and efficiently store data on RFID tags in XML format. We introduce a framework using cryptography that ensures data confidentiality and integrity; we employ multi-level encryption together with role-based access control on the data stored on an RFID tag. In the given framework, a user is assigned with a certain role and can only access

the part of data that she is authorized according to her role and the Access Control Policy (ACP). In addition, a more profound and accurate definition of simple and complex XACL (XML Access Control Policies) is given and a workable cryptographic solution is provided to handle complex policies. Furthermore, two different encryption methods are introduced to minimize the size of a file encrypted using XML encryption specifications. The research also extends the current technique of populating RFID tag memory with BIM (Building Information Model) database information in Facilities Management System (FMS) applications, by adding roles and different security levels. To explore the technical feasibility of the proposed approach, a case study in facilities management with different roles and security permissions has been implemented and tested at Concordia University. In this case study, we apply the proposed framework and encryption scheme to provide fine-grained access to data stored on RFID tags. To the best of our knowledge, it is the first work that addresses security issues in this new type of RFID-based distributed storage applications.

Acknowledgments

My greatest appreciation goes to my supervisors, Dr. Bo Zhu and Dr. Amin Hammad for their intellectual and personal support, encouragement and patience. Their advice and criticism was my most valuable asset during my studies. Overall, I feel very fortunate having the opportunity to know them and work with them. I would also like to acknowledge the cooperation of Mr. Ali Motamedi in executing the case study, Mr. Vitor Lima for developing the application.

Dedication

To my parents, Ramesh Chander Saini and Anil Kumari Saini and my brother Rajesh Saini who made all of this possible, for their endless encouragement and support.

Contents

List of Figures

List of Tables

List of Abbreviations

ACP	Access Control Policy
AECOO	Architects Engineers Constructors Owners and Operators
BIM	Building Information Model
BM	Building Management
DoS	Denial of Service
EAS	Electronic Article Surveillance
EH&S	Environmental Health and Safety Office
EPC	Electronic Product Code
FM	Facilities Management
FMS	Facilities Management System
GPS	Global Positioning System
GUI	Graphical User Interface
H&S	Health and Safety
HTML	Hypertext Markup Language

IAI	International Alliance for Interoperability
IBTTA	International Bridge Turnpike and Tunnel Association
IFC	Industry Foundation Classes
IFF	Identify Friend or Foe
ILR	Intelligent Long Range
INS	Indoor Navigation System
L/WAN	Local/Wide Area Network
NFPA	National Fire Protection Association
PCMCIA	Personal Computer Memory Card International Association
PDA	Personal Digital Assistant
RBAC	Role-Based Access Control
REP	RFID Enhancer Proxy
RFID	Radio Frequency Identification
ROI	Return on Investment
SGML	Standard Generalized Markup Language
SQL	Structured Query Language
W3C	World Wide Web Consortium
XACP	XML access control policy
XML	eXtensible Markup Language

Chapter 1

Introduction

1.1 General

In the past decade, there has been a boom in deploying Radio Frequency Identification (RFID) tags in different fields, e.g., electronic toll collection and contact-less passports [8]. Most protocols proposed so far focus on scenarios that RFID tags are used mainly for the purpose of identification and tracing. Recently, there rise interests of using RFID tags as distributed storage [18,27,46,52] so as to retrieve information related to a component, e.g., its status instead of only its identification, in real time, due to the benefit of convenient wireless management. However, such early work concentrates mainly on the basic functionality (e.g., storing and retrieving) of RFID tags, and security issues (e.g., access control to the data stored on an RFID tag) are largely ignored.

The wide usages of RFID have lead to extensive studies on RFID security, in particular authentication [9, 19, 38, 59, 61, 75, 76] and privacy [3, 25, 34, 37, 40–42, 62, 67] issues. In these security solutions, access control to the data stored on an RFID tag is enforced through authentication. In other words, an entity can retrieve all the information on the tag if it has already successfully passed the authentication process. In most cases, the tag only stores very limited information about the product (e.g., a unique product ID), which is not

sufficient to complete a task. Therefore, we have to refer to a centralized database to obtain more detailed information about the product to which the RFID tag is attached.

The centralized solution works well in many applications, e.g., books management in libraries [49] and patient and staff tracking in hospitals [48]. However, the effectiveness of this solution highly relies on the availability of an online centralized database. Thus, a major problem of this solution is the single-point-of-failure. In emergency-response scenarios, e.g., earthquakes or the outbreak of a fire, the communication with the centralized database may be totally unavailable. In addition, if an adversary could compromise the centralized database, all the confidential data stored on this database are disclosed or modified. Recent research shows that it is possible to compromise the centralized database using an RFID chip. Typical attacks include buffer overflow, code injection, and SQL injection [63, 64]. Moreover, the centralized solution may result in a delay in the completion of certain tasks, which is not suitable for scenarios that have real-time requirements.

Historically, the choice of the centralized solution is mainly due to the limited memory size of the first generation RFID tags. Nowadays, the memory size of an RFID tag (in particular an active tag) is large enough to hold the information on the tag itself. For example, IDENTEC SOLUTIONS's i-Q32T active UHF tag has a memory size of 32 KBytes [70]. The advance in RFID storage technologies makes it possible to use an RFID tag as a small local data storage, and thus greatly broadens the industrial usages of RFID tags, e.g., Facilities Management Systems (FMSs) and Indoor Navigation Systems (INSs) [54]. Moreover, centralized servers offer only a part of the information required to make effective facility maintenance decisions. They are not capable of providing accurate information about the state and structure of certain parts of a building throughout its lifecycle because they may fail to incorporate modifications made and additional data collected after the parts left the manufacturer, unless other organizations agree to provide this information to the central data repository. Moreover, with the global nature of todays supply chains, centralised

product databases are mostly impractical, since not all information about a single part can necessarily be kept by one company. In most cases it is practical to distribute the data among multiple databases [28].

A more detailed motivating example is given in Section 3.1. With sufficient memory capacity, it is possible to update RFID tag contents throughout the life-cycle as information about the part is collected or changed. On the other hand, in spite of the increase of the storage capacity of RFID tags, any method that can improve the cost/storage efficiency is still highly desirable. In terms of data management, therefore, we allow data used for different purposes/services to be stored on the same tag to maximize the usage of the RFID storage.

The choice of data location and storage entirely depends on the application. The following are some of the points which may justify the cause of storing data directly on the tags:

- Real-time decisions. In certain scenario (e.g., emergency like fire), decisions have to be made in real time. In this case, it might be inefficient to access necessary information about the building from a database held elsewhere on the network. Here, storing data on the tag could be a more efficient method as real-time decisions require real-time availability of information.

- Real-time data capture. There are many situations where data need to be captured and recorded in real-time throughout the lifecycle of the building/tag. An example of such a case is monitoring of temperature variations in different parts of a building. Here, temperature sensors might be attached to RFID tags and variations/abnormalities could be recorded directly on to the tag itself.

- Data access/update at remote locations. Another reason why data might be kept on the tag is if data need to be available immediately in a place where access to a

networked database is not available. For example, wireless signals are not available in basements.

- Frequent data access. Throughout the lifecycle (e.g., manufacturing, shipping, transportation, operation and inspection) of a building, there are various decisions that need to be taken, which in turn use different sets of information about the building. Some information would need to be accessed more frequently than others. In situations where the frequency of data access is high and the cost of data retrieval and transmission is greater than the cost of the writable tag, it is advisable to store data on the tag.

The distributed storage of data on RFID tags can provide many benefits, such as real-time data access and emergency response. As a trade-off, it also bring certain potential risks and challenges. Since the actual data are stored on tags instead of a back-end database, a major concern will be data protection, including both confidentiality and integrity.

Current solutions based on authentication are vulnerable to physical attacks, since RFID tags are installed at open areas. Although there is little formal research on the effectiveness of physical attacks against RFIDs, as a type of smart cards, current physical attacks against smart cards [2,24,83] are also applicable to RFID tags. As a result, data should be encrypted not only during the communication between an RFID tag and an RFID reader but also when they are stored on the tag.

In our Facilities Management (FM) and emergency response applications (explained in chapter 4), different parts of the data may be accessible only by a specific set of users, and sometimes the data stored on a tag may belong to multiple owners, who do not have full trusts in each other. Therefore, a naïve solution that encrypts all the data with a single key does not work. A possible solution is to assign each specific set of users (or say *role*) a shared key and then encrypt all the data accessible by a role with its corresponding key. However, this method is workable only when the data accessible by any pair of roles are

totally disjointed, and is unsuitable for the scenarios in which the data accessible by a pair of roles are overlapped[1]. In the latter, multiple ciphertexts corresponding to the overlapped data are generated and stored. Apparently, it leads to poor storage efficiency. Moreover, the extension of generating a specific key for the overlapped data is infeasible due to the key management issue.

1.2 Research Objectives and Contributions

The objectives of this research are: (1) To review current research on the security of RFID technology; (2) To probe the idea of providing security on the RFID tags within existing constrains of limited computation capability; (3) To define simple and complex XML Access Control Policy (XACP) and generate an algorithm that determines the exact type of Access Control Policy (ACP); (4) To investigate cryptographic solution to handle complex policies; (5) To demonstrate the effect of different encryption modes on storage efficiency; and (6) To demonstrate the feasibility of the proposed approach through a real world case study.

To address these challenges, in this thesis, we propose the first framework that provides efficient data protection in *RFID-based Distributed Storage* (**RDS**) applications. Our contributions are:

- To ensure data confidentiality and integrity, we employ multi-layer encryption together with role-based access control on the data stored on an RFID tag. In our scheme, although any user can download the data stored on a tag, the protection of the data or a certain part of the data relies on the key(s) that is/are assigned to the user instead of the on-tag authentication mechanism. A user can only access the part of data that she/he is authorized according to her/his role and the ACP.

[1] In a special case, the data accessible by one role is contained in the data accessible by another role.

- Previous definitions of simple XACP and complex XACP are inaccurate. In this paper, we redefine these two terms in terms of data publishing.
- We design an algorithm that determine the exact type of the ACP, simple or complex. This output can be used as a guidance for the selection of a storage-efficient encryption method.
- Notice that previous cryptography-based solutions to complex XACP are considered problematic [10], we propose a solution to this issue.
- We present different XML encryption modes, apart from standard XML encryption specification, to enhance storage efficiency.

In this thesis, we focus on the case that the data stored on an RFID tag are represented in XML [77], although the proposed framework and algorithms can be readily extended to support other types of data representations. In addition, although we focus on the RFID-based distributed storage applications in this paper, the proposed framework and algorithms can be used for access control on data publishing in general, in particular when there is a concern about the storage efficiency.

1.3 Thesis Organization

The thesis is organized as follows:

Chapter 2 Literature Review: This chapter reviews the major technologies and standards that are used in the research. Literature review comprises the history of RFID technology, its components and details about RFID applications and previous security solutions. Facilities Management Systems and Indoor Navigation Systems are briefly covered in this chapter. This chapter also provides introduction to XML technology and some of

its features such as, XML Access control and XML Encryption. Moreover, XML research problem evolving around access control policies and previous solutions are discussed.

Chapter 3 Proposed System Model and Framework: In this chapter, the proposed security framework for securing data stored on RFID tags using XML super encryption and role based XML access control is elaborated. This chapter includes conceptual and interaction design of the system. All the steps involved in encryption and decryption are explained explicitly. This chapter also goes through the procedure of key generation for a role and gives a brief introduction to multi-layer encryption. In order to elaborate more on the applicability of the introduced approach, experimental results are provided that give the impact of different XML encryption modes on storage efficiency.

Chapter 4 Case Study: In this chapter, the proposed approach is demonstrated by means of a case study. In the case study, the approach is used to facilitate a secure communication protocol for a facilities management and emergency response system. Conceptual and software design along with a prototype software are explained.

Chapter 5 Conclusions and Future Work: This chapter summarizes the present research work, highlights its contributions, and suggests recommendations for future research.

Chapter 2

Literature Review

2.1 Radio Frequency Identification

RFID, is a broad term that contains several information and communication technologies that use radio communication to uniquely identify objects [6, 31] by storing a certain amount of data such as the product identification, price or manufacturing date in RFID tag's memory. Unique identification with RFID becomes explicit: the object, location or individual is assigned a unique identifier code contained within an RFID tag, which is in some way attached to or embedded in the target. An RFID reader is used to search for tags, when a reader receives a signal that a tag is present in its vicinity, it retrieves data from tag wirelessly providing the ability to process large amount of multiple data set simultaneously for a particular application.

2.1.1 RFID Architecture and Components

The typical RFID system consists of a reader (also called beacon), tags (also called transponder or chip), antenna and backend server that receives and processes the information that the reader collects from the tags. The emission of radio signals by the reader's antenna

activates a tag to read or write data from/to it. The reader is responsible for transmitting adequate energy to power up and communicate with the tag to request and receive the data. The range of reader's radio waves depends on its power and radio frequency and can activate tags at a distance up to 100 feet or more. When RFID tags come under the electromagnetic zone generated by reader's antenna, they detect the activation signal and respond by sending stored data in the form of electromagnetic waves. The reader then decodes the received data which is encoded in the integrated circuit of the tag and passes them to the backend server for processing [14,23]. A conventional RFID system is shown in Figure 2.1

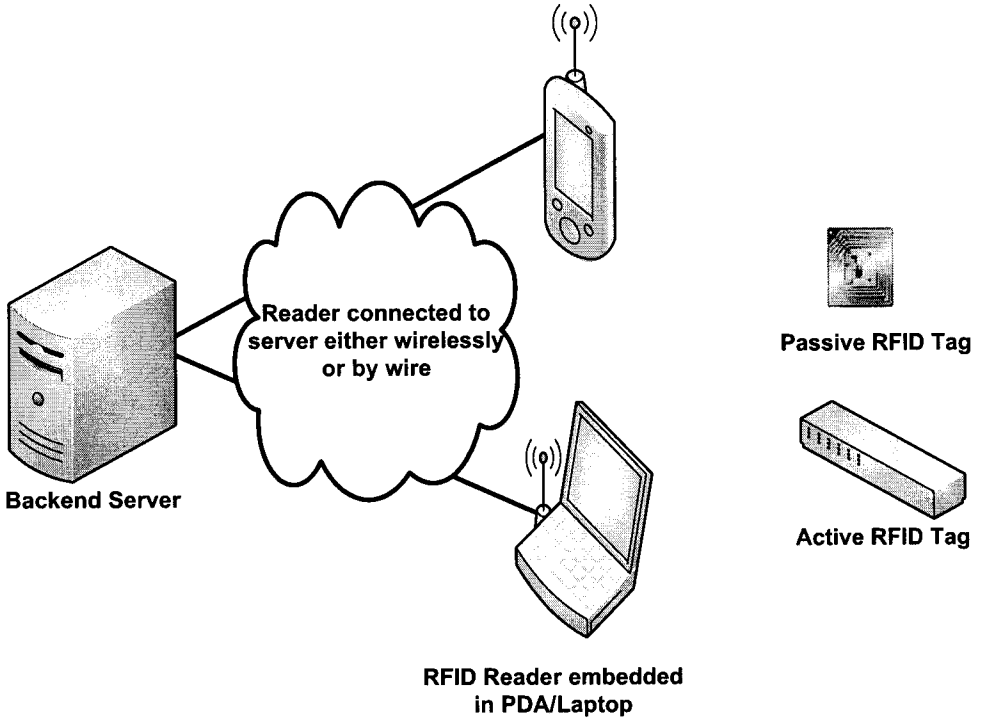


Figure 2.1: Conventional RFID System

Tag

RFID tag is a microchip joined with antenna in a compact package. It comes under the category of transponders, which is a combination of transmitter and receiver and is designed to

receive a specific radio signal and automatically transmit a reply. A tag can send from simple reply signal to single digit or multiple set of strings of letters. In the near future, RFID tags may be able to do complex calculations like encryption and decryption. In general, RFID tag contains encoding/decoding circuitry, power supply, antenna, communication control and memory [74].

Tags can be further classified into three categories: passive, semi-passive and active.

- *Passive:* The tags that do not carry battery or any other power source and depend entirely on the reader for their power fall in this category. These tags contain a resonant circuit capable of absorbing power from reader's antenna [74]. To get data from passive tags, the reader's antenna should be in close vicinity of the tag and with higher power as compare to active tags. Passive tags are much lighter and less expensive than active tags, and virtually offer unlimited operational lifetime. Figure 2.2 shows high level view of reader and passive tag communication.

- *Semi-Active:* Unlike passive tags, semi-active tags have battery of their own to provide power to the memory chip for on-tag calculations only. Semi-active tags have bigger memory size than passive tags. Though semi-active tags have battery, still they rely on power from reader to start communication.

- *Active:* The tags have their own power source, used for both on-tag calculation and communication with reader. As these tags do not depend on reader's power, they are able to transmit and receive electromagnetic radio signals over a long range of distance and support more reliable communication. Further, active tags can operate very well in challenging environments like places with significant radio frequency pollution caused by electric machinery. It is easy to extend active tags with additional sensing capability, like temperature sensors [65, 70]. Active tags are usually bigger and more expensive than passive and semi-active ones and have a limited operational

life which may yield a maximum of 10 years, depending on operating temperature and battery type [23]. Active tags stop operating when their battery expires [65].

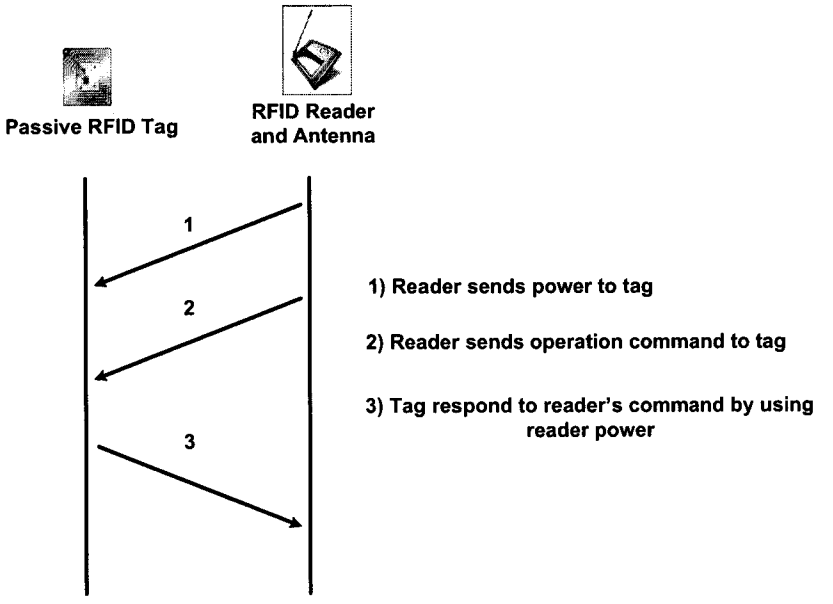


Figure 2.2: High-Level View of Reader and Passive Tag Communication [65]

Reader

The reader retrieves the information from the RFID tag. The reader may be self-contained and record the information internally, or it may also be a part of a Local/Wide Area Network (L/WAN). Readers send data to a LAN or other systems by using a data interface such as Ethernet or serial RS-232. Other parts that a reader typically contains are a system interface such as an RS-232 serial port or Ethernet jack, cryptographic encoding and decoding circuitry, a power supply or battery, and communications control circuits. Readers can be of different sizes from postage stamp-sized to large devices with panels that are several feet wide and high [74]. Figure 2.3 describes the communication process between readers and tags.

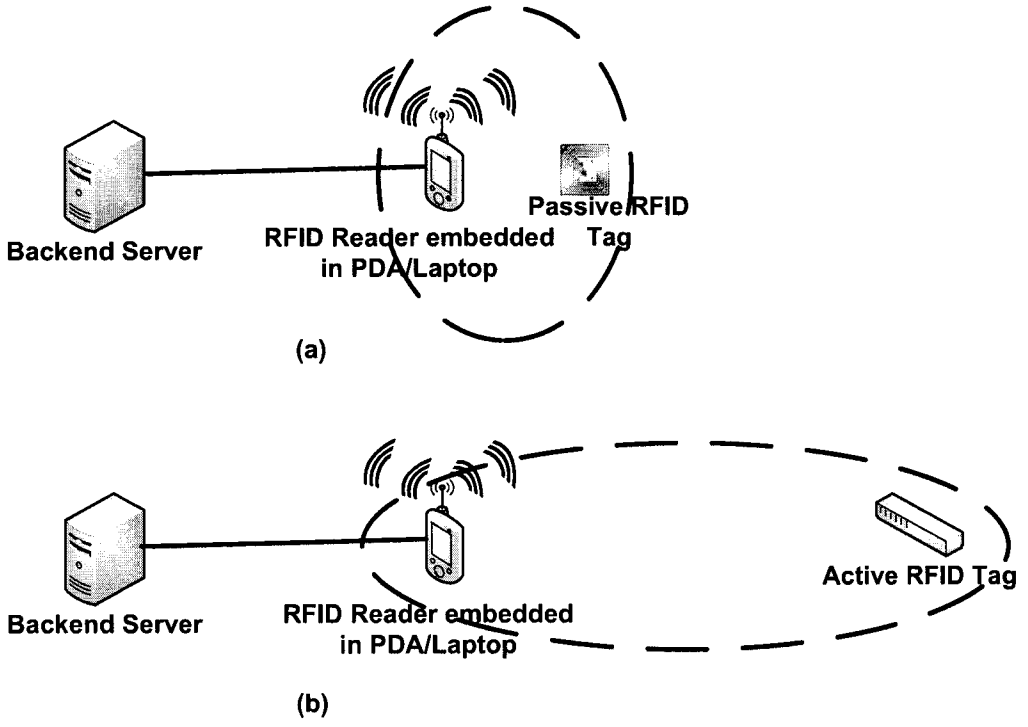


Figure 2.3: (a) Data Exchange Occurs at Close Distance Between Antenna and Passive Tag (b) Data Exchange Occurs at Longer Distance Between Antenna and Active Tag [74]

Antenna

The antenna can be an integral part of the reader, or it can be a separate device. Handheld units are a combination reader/antenna, while larger systems usually separate the antennas from the reader.

Middleware

Middleware is a very important component in conventional RFID systems. It manages the readers and data transmitted from the tags, and then passes to the backend server. Middleware is a software runs on ordinary computers or servers, which contains the logic of the RFID application, and a backend database system for storing information about the tags. Middleware is placed in between the reader and the backend, and manages the flow

of information. It also performs functions such as basic filtering and reader integration and control. With the passage of time, the middleware is supposed to improve its features and expand management capabilities and extend data management options [74]. Middleware is prone to many security attacks like Man in the Middle attack. The backend can be any standard database server such as Oracle, SQL, etc. and suffers from conventional database security threats.

2.1.2 Brief History of RFID Technology

RFID is a combination of radar and radio broadcast technology. The starting of RFID technology can be traced back to development carried out on radar technology during World War II. Identify Friend or Foe system (IFF), first introduced in WWII is considered as the initial phase of today's RFID. IFF systems were further developed in the 1950s and nowadays are in common use in civil and military aviation [13, 65].

Harry Stockman, published a paper entitled "Communication by Means of Reflected Power" [73]. In the late 1960s, two companies called Sensormatic and Checkpoint together with another company called Knogo, developed the Electronic Article Surveillance (EAS) tags. These are 1 bit tags that are attached to items in order to prevent and detect the theft of merchandise, EAS tags are very cheap and are still in use today [45].

Commercialization of RFID was not started until the 1970s. In the early 1970s, RFID was used for access control, following Charles Walton's patent of a tag used to unlock a door without a key [36]. During the same period, large companies such as Raytheon and RCA developed electronic identification systems. The Los Alamos Scientific Laboratory, the International Bridge Turnpike and Tunnel Association (IBTTA) and the United States Federal Highway Administration organized a conference on RFID in 1973 which concluded that there was no national interest in the development of a standard for vehicle

identification [14]. This decision lead to the development of a range of RFID related systems. In 1978, R.J. King wrote a book about microwave homodyne techniques. This book has been used as the basis for the development of the theory and practice which are used in backscatter RFID systems [45].

In the beginning of 1980s, RFID found applications in cattle tagging and railroad freight tracking. At the end of the 1980s, with the rapid miniaturization of electronics, which offered at the same time lower cost and higher performance and capacity, RFID technology became commonplace and found a variety of applications. One particular area of major growth has been the use of passive inductive tags to develop a variety of contact-less smart cards which have found popular applications, especially in access control and ticketing [65].

During 1990s, a number of American states, such as Kansas and Georgia, adopted a traffic management system which was based on the use of readers that could detect RFID tags. Europe also followed the American foot steps in traffic management. Texas Instruments developed the TIRIS system which was used in applications related to vehicle access. European companies, such as Alcatel, Bosch and Phillips spin-off companies, such as Combitech, Tagmaster and Baumer were involved in the development of a pan-European standard for tolling applications. These companies helped develop a common standard for electronic tolling.

At the beginning of the 2000s, RFID came into prominence due to its unique capability to automatically identify tagged entities at potentially very low cost. At the same time, the internet has been established as the primary infrastructure for the operational deployment of network services that could complement well the advantages of RFID. In 2004, EPCGlobal [15] released the second generation standard for Electronic Product Code (EPC) tags. With the passage of time, tags are shrinking in size and cost, while gaining new capabilities.

2.1.3 RFID Applications

RFID is a versatile technology, capable of being used in vast number of fields. Huge success of RFID in supply chains and replacement of barcodes, have overshadowed how extensively and successfully RFID is used in other contexts. RFID has been identified as one of the ten greatest contributory technologies of the 21st century. This technology has found a rapidly growing market; in 2008 the RFID market approached US$ 4 billion, growing approximately 35% from 2007. RFID market is estimated to exceed US$ 4.4 billion in 2009 an 11% growth rate over 2008 [60]. An increasing variety of enterprises are employing RFID to improve their efficiency of operations and to gain a competitive advantage [7]. The following are the some of the RFID uses [74]: supply chains (including wholesale and retail inventory and materials management), item-level tagging of consumer goods on retail shelves, toll payment systems, smart cards, contact-less payment systems, asset tracking, automobile keyless start systems, sports, ticketing, access control, luggage tracking, passports and border control, libraries and building management. Apart from the above mentioned applications, RFID has a great potential in FMS and emerged as a strong contender for INS.

Facilities Management Systems (FMSs)

The Construction Industry Institute [44] pointed out a number of potential application areas for RFID technology in the sector. These included component tracking and locating, inventory management, equipment monitoring, progress management, facilities and maintenance management, tool tracking, material management and quality control [17, 43, 71]. However, the cost of implementing RFID solutions has remained high as each of the above-mentioned applications is designed for only one specific stage of the facility lifecycle to serve the needs of only one of the stakeholders in a fragmented fashion, i.e., Architects, Engineers, Constructors, Owners and Operators (AECOO). Further, using different tags at

different stages eliminates the chance of using shared resources among the stakeholders causing duplication of efforts and resources.

The AECOO industry is highly fragmented in nature. Thus, it involves bringing together multi-disciplines and different parties in a project that requires a tremendous amount of coordination. There is an evident need for a standard information transfer model between different software applications used in the AECOO industry. The Building Information Model (BIM) has been developed in order to tackle the problems related to interoperability and information integration by providing effective management, sharing and exchange of a building information through its entire lifecycle [35].

According to [26], BIM is a data-rich, object oriented, intelligent and parametric digital representation of facilities. [58] described the scope of BIM within the following relationships: (1) BIM as a product or intelligent digital representation of data about a capital facility, (2) BIM as a collaborative process which covers business drivers, automated process capabilities, and open information standards use for information sustainability and fidelity, and (3) BIM as a facility lifecycle management tool of well understood information exchanges, workflows, and procedures which stakeholders use throughout the building lifecycle as a repeatable, verifiable, transparent, and sustainable information based environment. BIM acts as an enabler of interoperability and is a facilitator of data sharing and exchange between software applications. Furthermore, BIM is extensible, open and vendor neutral [35]. Figures 2.4 and 2.5 conceptually show how BIM data chunks are stored on tags attached to the building components. While the information is centrally stored in the BIM database, software applications copy the necessary information from the database to the memory space on the tags [54].

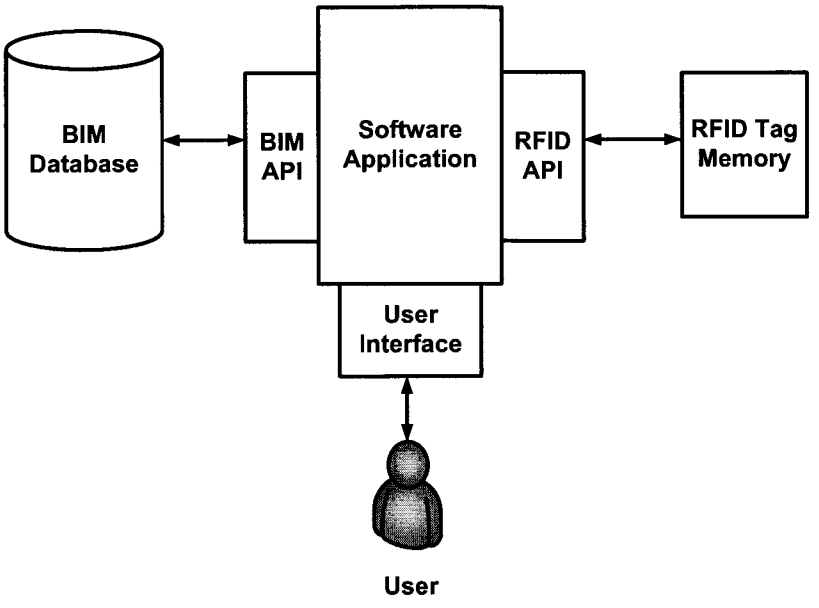


Figure 2.4: Conceptual System Interaction Design [54]

Indoor Navigation Systems (INSs)

In the last few years many systems have addressed the problem of automatic positioning. Triangulation, scene analysis, and proximity are the three principal techniques for automatic location-sensing [32]. The most common example is Global Positioning System (GPS) [33], as it is satellite dependent, it is not able to accurately identify the position of objects or people inside building.

Based on the idea of GPS a much varied set of alternative INSs have been developed over the years, based in technologies such as infrared [82], ultrasonic/sonic signals [29], artificial vision and now with RFID [4,30]. While, position-sensing was not the primary task of RFID, recent studies showed that no-contact, non-line-of-site nature, long transmission-range and low cost are some of the advantages of RFID over other technologies. Indoor localization technology is becoming more sophisticated and affordable day by day. Many indoor location-aware applications such as emergency management in hospitals [48], airport security [50], navigation of autonomous vehicles, and assistance to people with disabilities,

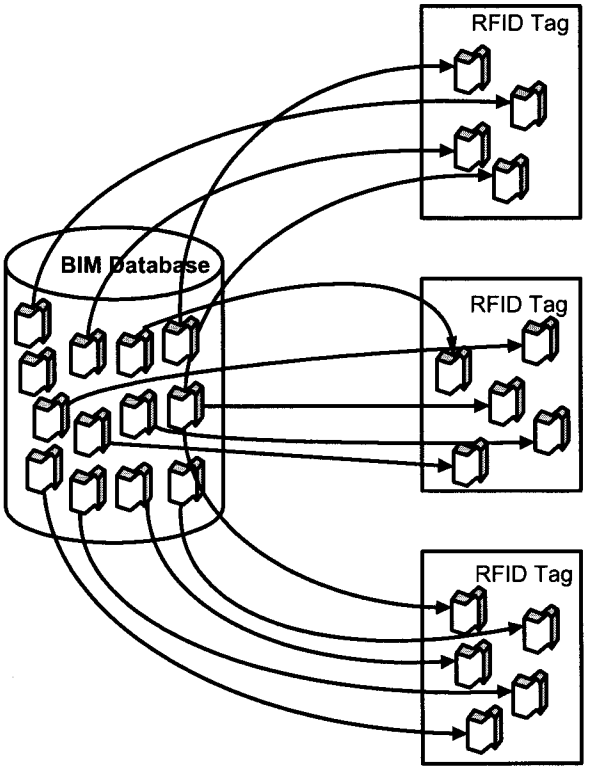


Figure 2.5: Conceptual BIM-Tag Data Relationship [54]

etc., require knowledge of position of mobile objects or persons for their operation.

There have been many positioning systems based on two prototypes. In the first prototype, RFID readers are placed at certain fixed locations. Each reader defines a certain range in which it can detect RFID tags because of pre-determined power level. The whole region is further divided into sub-regions by placing the readers at known positions. Each sub-region can be uniquely identified by the subset of readers that cover that sub-region. An RFID tag is associated with that known sub-region, which detects it, based on the subset of readers. The accuracy of this approach is then determined by the number of readers required, placement of readers and the power level of each reader. There has been many variations is in this prototype to improve its efficiency such as [57, 68].

Another prototype is just opposite of previous one, where tags are fixed to the building's

infrastructure and the reader is the mobile agent, whose relative position needs to be estimated. The location information stored on the tags helps estimate the approximate position of the RFID reader relative to the positions of the tags detected in is immediate vicinity. Deployment information assessable from the tags permits estimation of the reader position with respect to the map where they belong. This application is of significant importance during emergency situations, where an agent requires real-time localization and possible path planning towards a certain destination (e.g. an exit during a fire) [22].

Above mentioned prototypes have one common limitation. The approximate position calculated is always with respect to some part of a map where the tag or reader is placed, but the map itself may not be available at runtime. The User should have the associated map or floor plan of the building in his hand held device before using this positioning system. Also, due to limited memory it is not possible to store whole floor plan on RFID tags. The problem becomes further difficult as every building has a set of floor plans and storing all these plans at one place would not be very cost efficient. To make the proper use of increasing active RFID tag memory, we propose to divide a floor plan into small sections and store them on different tags. This information helps the navigator to build the whole floor plan as he/she moves along without any prior knowledge of building map.

2.1.4 RFID Security Vulnerabilities

Due to large number of advantages over other technologies, RFID is becoming more popular and expected to replace current technologies, such as barcodes. However, privacy protection, authentication issues and other security vulnerabilities which make it an easy target for malicious attacks, lead to growing concern among RFID users and vendors. Attack on RFID can be on the entire system, or it can be on any part of the overall system. This section discusses various attacks that can occur on RFID systems and applications and previous research work to tackle these attacks.

Attack on Radio Frequency

Spoofing attack provides false information that appears valid and accepted by the system. Broadcasting an incorrect EPC number over the air when a valid number was expected is an example of spoofing in RFID system.

Replay attack, a valid RFID signal is intercepted and its data is recorded; these data are later transmitted to a reader where they are played back.

Denial of Service (DoS) attacks, also known as flood attacks, take place when a signal is flooded with more data than it can handle. A variation on this is RF jamming.

Attack on Tag Data

Certain programs like RF Dump scans for RFID tags via reader attached to the serial port of a computer. When the reader recognizes a card, the program presents the card data in a spreadsheet-like format on the screen. The user can then enter or change data and reflect those changes on the tag. RF Dump also makes sure that the data written is the correct length for the tag's fields, by either padding zeros or truncating extra digits as needed. A Personal Digital Assistant (PDA) program called RF Dump-PDA is available for use on PDAs such as the Hewlett-Packard iPAQ Pocket PC. RF Dump-PDA is written in Perl, and will run on Pocket PCs running the Linux operating system. Using a PDA and RF Dump-PDA, a thief can walk through a store and change the data on items [74].

Attack on Backend Server

The Database is a very crucial part of the RFID system. It may hold valuable information like trade secret or customer's credit card number, manipulation or theft of which can lead to dire consequences, especially, in applications such as hospital's inventory, where a change of one letter involving a patient's blood type could put patient's life at risk and building management system where replacing the material used in construction could result

in building collapse.

Structured Query Language (SQL) injection and buffer overflow attacks, are fairly new ideas. RFID interface can be used to insert information into the database, unless proper checking systems are used to guarantee that only legitimate tags are trusted. Recently a group in Netherlands created RFID viruses and worms that fit a malicious program (malware) onto the memory area of a programmable RFID chip (i.e., a tag). When the chip was queried by the reader, the malware passed from the chip to the backend database, from where the malware could be passed to other tags or used to carry out malevolent actions. The exploits employed, including SQL and buffer overflow attacks, are generally used against servers [74]. Other drawback of backend server is a potential single point of failure, so it needs to be redundant, distributed and secure.

2.1.5 Previous Solutions

In the majority of the applications mentioned above, passive tags are used mainly because of their less cost and virtually endless life. The two main security concerns of passive tags in their current format are privacy and authentication. Privacy is further categorized into two problems of clandestine tracking and inventorying [39]. As RFID tags respond to any compatible reader without distinction, the matter of a nearby RFID reader being able to scan any RFID tag is dig into by clandestine tracking. On the other hand clandestine inventorying is about getting sensitive information from the tags. Authentication is another big security concern, it can be done in two ways. By authenticating a reader to a tag, and by authenticating a tag to a reader. Extensive research in RFID security has been carried out in both privacy and authentication.

Privacy Solutions

The simplest method to protect privacy is by preventing RFID tags from communicating by enclosing them in Faraday cage [39] such as a metallic sheet.

The Blocker Tag [40] was designed to block selectively by simulating only selected subsets of ID codes, such as those by a particular manufacturer, or those in a designated privacy zone. The Blocker Tag can be bypassed easily if a reader does not follow the singulation protocol.

EPCglobal chip designs address the privacy problem by allowing an RFID tag to be *killed* [16]. Additional processing capability is required in the tag to protect the password against the unauthorized use that eventually kills the tag.

In Clipped tags [42], consumers can physically separate the body (chip) from the head (antenna) in an intuitive way. Such a separation provides visual confirmation the tag has been deactivated. However, a physical contact channel may be used later to reactivate it. Such a reactivation would require deliberate actions on the part of the owner of the RFID tag to permit the reactivation to take place.

RFID *Enhancer Proxy* (REP) [41] assumes the identities of tags and simulates them in the presence of reading devices by continuously relabeling their IDs. The REP suffers from a number of shortcomings such as corruption of tag data, tag-to-REP de-synchronization and difficulty in tag release that are attributed to the fact that tag identities need to be partially generated by the tag and match portions of its true ID. Some of other works like [25, 34] also deals with privacy issues.

Hash-Locking [84], is a method that uses one-way hash function to generate a metaID that obscures the tag's original ID while providing an index to find the tag's ID in the database. Since there is no dynamic mechanism to randomize the metaID, it acts as an identifier in plaintext. This makes protocol vulnerable to counterfeit attack and replay attack.

Authenticaiton Solutions

The RFID Guardian [62] looks for, records, and displays all RFID tags and scans in the vicinity, manages RFID keys, authenticates nearby RFID readers, and blocks attempted accesses to the user's RFID tags from unauthorized readers. RFID Guardian acts as an intermediary between tags and readers and must always be active in protecting tag responses from unauthorized read attempts. It has to either allow reader queries, appropriately reissuing queries in encrypted form, or actively block tag answers. Thus if the Guardian fails, security is lost. Furthermore, it does not deal with such issues as tag acquisition and ownership transfer.

Other authentication protocols such as [19, 37, 38, 41, 42, 59, 62, 75, 76] mainly rely on encryption, nonce or challenge-response. All these solutions require tag to do some kind of computation like random number generation and encryption/decryption. Nonce is used to ensure the communication is fresh and cryptography is needed to secure the communication. These solutions to authentication problems are beyond the current state of the art of RFID because present tags lack the computing capability to do even simple cryptographic calculation. The cryptographic capabilities necessary for vigorous security will require an increase of orders of magnitude in circuit complexity. In most protocols, the discussions focus on scenarios that RFID tags are used mainly for tracing or identification, and the access control of data stored on RFID is enforced through authentication.

Unfortunately, such mechanisms are unsuitable for the new type of applications like indoor navigation and facilities management, in which the actual data are stored and distributed on the RFID tags instead of a back-end database. To the best of our knowledge, this research is the first work that investigates security issues, in particular access control, in this new type of applications.

2.2 eXtensible Markup Language (XML)

The eXtensible Markup Language (XML) [77], recommended by World Wide Web Consortium (W3C) is a standard to describe, store and exchange information on the Internet and through other means. XML is a restricted subset of Standard Generalized Markup Language (SGML). XML is similar to Hypertext Markup Language (HTML). Both XML and HTML consist of markup symbols to illustrate the contents of a page or file. The XML specification defines a syntax for creating markup. Elements, attributes and other structures that are used to label documents and data in a manner that makes sense to computer programs and even humans are part of markup. XML gives power to users to define their own tags which makes XML self-descriptive. XML contains self-defined data in document format; so as syntax it is platform independent. Some of the key features of XML are flexibility, open standard, enhanced scalability and compression. Also, the order in which data appears is not important in XML. In short, XML is designed for sharing information easily via a non-proprietary format over different channels. Figure 2.6 shows a sample XML file with custom-defined tags for product inspection in a FMS.

2.2.1 XML Encryption

XML encryption [79] is a W3C standard for encrypting XML documents. It defines a course of action of encrypting and decrypting digital XML contents, using specific syntax and algorithms. The basic concept of cryptography remains mostly the same, the difference comes in representing and exchanging encrypted XML contents. XML encryption specification includes the standard syntax for representing the encrypted contents within XML, along with information needed to decrypt the contents on the receiving side. The encryption method consists of taking an element from an XML document, encrypting it and all its children and then replacing the original XML content with encrypted XML such that the newly generated document remains well formed. There are three ways of XML

```
<?xml version="1.0" standalone="no"?>
<Building-Management Name="EV Building" xmlns=
'http://example.ca/xml/nmsp/building'>
    <Inspection>
        <Hazard-Material>
            <Room Number="">123</Room>
            <Type>XYZ</Type>
        </Hazard-Material>
        <Product>
            <ID>A12</ID>
            <Product-Specification>....</Product-Specification>
            <Status>UP</Status>
        </Product>
        <History>
            <Comments>....</Comments>
            <Date>....</Date>
            <Inspector>....</Inspector>
        </History>
    </Inspection>
</Building-Management>
```

Figure 2.6: XML Example File for Product Inspection in Facilities Management System

encryption [69]:

- *Using symmetric encryption only*: Only one session key is used to encrypt and decrypt. The key itself is not stored with the encrypted document.

- *Using combination of asymmetric and symmetric*: In this method a session key is used to encrypt the data and an asymmetric key to protect the session key. Both, encrypted session key and data are stored together in XML document. The public asymmetric key is used to encrypt the session key while the private asymmetric key is used to decrypt the key.

- *Using X.509 certificate*: This approach uses X.509 certificate as the symmetrical key. X.509 certificate are provided by a third party vendor.

After encryption either the whole element is replaced with an element named EncryptedData or just the data in the element is replaced and its name remains readable in the encrypted document. As shown in Figure 2.6, Inspection element contains child element called Hazard-Material which may contain some confidential information regarding chemicals stored or used in a room. If data owner does not want every user to know about this information, it would be more appropriate to replace the whole element. Usually it depends on what the data is and how much information is necessary to give away. Figure 2.7 shows XML document from Figure 2.6 encrypted with a public key of recipient using asymmetric approach where Hazard-Material element and its child elements are replaced with EncryptedData

Super-Encryption

An XML document may contain zero or more EncryptedData elements. EncryptedData cannot be the parent or child of another EncryptedData element. However, the actual data encrypted can be anything, including EncryptedData and EncryptedKey elements (i.e., super-encryption). During super-encryption of an EncryptedData or EncryptedKey element, the entire element must be encrypted. Encrypting only the content of these elements, or encrypting selected child elements is an invalid instance under the provided schema [79]. For example, consider Figure 2.8.

A valid super-encryption of EncryptedData[Id='ED1'] in Figure 2.8 is shown in Figure 2.9, where the CipherValue content of 'newEncryptedData' is the base64 encoding of the encrypted octet sequence resulting from encrypting the EncryptedData element with Id='ED1'.

```
<?xml version="1.0" standalone="no"?>
<Building-Management Name="EV Building"
xmlns='http://example.ca/xml/nmsp/building'>
    <Inspection>
        <EncryptedData Type='http://www.w3.org/2001/04/xmlenc#Element'
            xmlns='http://www.w3.org/2001/04/xmlenc#'>
            <EncryptionMethod
        Algorithm='http://www.w3.org/2001/04/xmlenc#tripledes-cbc'/>
            <KeyInfo xmlns='http://www.w3.org/2000/09/xmldsig#'>
            <EncryptedKey xmlns='http://www.w3.org/2001/04/xmlenc#'>
            <EncryptionMethod Algorithm='http://www.w3.org/2001/04/xmlenc#rsa-
        1_5'/>
            <KeyInfo xmlns='http://www.w3.org/2000/09/xmldsig#'>
                <KeyName>Rakesh Saini</KeyName>
            </KeyInfo>
            <CipherData>
                <CipherValue>UfREw13Az467G7...</CipherValue>
            </CipherData>
            </EncryptedKey>
            </KeyInfo>
            <CipherData>
                <CipherValue>dhYj4FtcQ24ftI...</CipherValue>
            </CipherData>
        </EncryptedData>
        <Product>
            <ID>A12</ID>
            <Product-Specification>....</Product-Specification>
            <Status>UP</Status>
        </Product>
        <History>
            <Comments>....</Comments>
            <Date>....</Date>
            <Inspector>....</Inspector>
        </History>
    </Inspection>
</Building-Management>
```

Figure 2.7: Encrypted XML File Using Asymmetric Approach

2.2.2 XML Access Control

In the past few years, XML expanded its sphere from being a standard for web document to become broadly popular as a data encoding format. Whereas conventional methods employed server-side techniques such as security views and query re-writing, there have been some doubts on the scalability of these approaches to manage applications involving large number of users and complex ACP. Another technique called secure publishing has been explored, in which the owner of the data source publishes data for public user usually

```
<?xml version="1.0" standalone="no"?>
<Building-Management Name="EV Building" xmlns=
'http://example.ca/xml/nmsp/building'>
    <Inspection>
        <EncryptedData Id='ED1' xmlns='http://www.w3.org/2001/04/xmlenc#'
        Type='http://www.w3.org/2001/04/xmlenc#Element'>
            <CipherData>
                <CipherValue>originalEncryptedData</CipherValue>
            </CipherData>
        </EncryptedData>
        <Product>
            <ID>A12</ID>
            <Product-Specification>....</Product-Specification>
            <Status>UP</Status>
        </Product>
    </Inspection>
</Building-Management>
```

Figure 2.8: Encrypted XML File Using Symmetric Approach

on insecure channels. In general, the data owner may have some sensitive information that should be available to selected users only.

The need of distribution and sharing of information leads to the importance of precise and secure access of XML data. Certain users are allowed to access specific parts of information stored in an XML document while preserving confidentiality of the rest of the document. Hence, access control over XML data is required to ensure that only authorized users have access to the parts of the data they are allowed to. Otherwise, authorized users have no access to any part of the document. The ACP defines how the document appears for different users. ACP provides XML with a fine-grained access control mechanism that enables the user to securely browse XML documents. ACP is used to specify an object-subject relation in the context of a particular XML document. Where, subject is a user or role and object is as fine as single element or a set of elements within the document.

The required access control is enforced by encrypting regions of the XML document using cryptographic keys. All authorized users get the set of keys corresponding to the

```
<?xml version="1.0" standalone="no"?>
<Building-Management Name="EV Building" xmlns=
'http://example.ca/xml/nmsp/building'>
   <Inspection>
      <EncryptedData Id='ED2'
   xmlns='http://www.w3.org/2001/04/xmlenc#'
   Type='http://www.w3.org/2001/04/xmlenc#Element'>
         <CipherData>
      <CipherValue>newEncryptedData</CipherValue>
         </CipherData>
      </EncryptedData>
   </Inspection>
</Building-Management>
```

Figure 2.9: Super Encryption : XML File Encrypted Twice Using Symmetric Approach

assigned permissions under the ACP. An ACP enforces them to decrypt only parts of the document consisting of element nodes they have been granted access to.

XPath [78] or XQurery [81] is used to extract different parts of the XML document as defined in ACP. An ACP can be defined as a sequence of XPath filter [80] expressions, which are combined using operation set intersection, subtraction and union. The set of nodes which must be hidden from a particular user are represented by the set *Subtract*. For example, consider Figure 2.6 and let an ACP states that a user is denied access to XML tag Hazard-Material and all its children. The given policy can be specified as:

```
\textbf{<dsig-xpath:XPath Filter="intersect">
// Inspection
</dsig-xpath:XPath>
<dsig-xpath:XPath Filter="subtract">
// Inspection// Hazard-Material
<dsig-xpath:XPath>}
```

Role-Based Access Control (RBAC) [20] approach has been proposed in recent XML access control solutions [10, 85]. RBAC has become preferred choice of defining and implementing access control since its introduction in 1995. The idea of RBAC is very simple, different roles exist in the system and permissions are assigned to roles instead of individual users, it then manage access to information with respect to these roles. To access information, a user must be a member of a predefined role and inherits the authorization privileges of that role.

2.2.3 Limitations of XML RBAC

One naïve way to implement ACP is using super-encryption, in which every single node is encrypted with a unique key but this leads to drastic increase in size of a document many folds. Furthermore, this implementation may end up assigning many keys.

A research topic that is tightly relevant to our work is secure data publishing of XML documents over the Internet. Many XML access control models have been proposed so far [5, 10, 12, 51, 55]. There are two major barriers of directly applying these solutions to the RFID-based distributed storage applications. One is the limitation on the storage. For example, the work done by Miklau and Suciu [51] generates many meta-nodes which increases the size of original document. Another weakness of this work is that users may have to maintain multiple keys. The other barrier is the absence of an online trusted server that enforces the XACP. In addition, previous cryptographic-based solution [10] focuses on scenarios with simple XACPs, and there is a lack of detailed discussions and feasible solutions, given that there is no online trusted server, for scenarios with complex XACPs.

2.3 Summary

In this chapter various technologies, standards and applications related to RFID and XML were reviewed. The literature showed that RFID technology has the potential to facilitate several distributed storage applications. The current security solutions available are not capable of providing confidentially and integrity on the data stored on RFID tags in these applications. On the other hand, XML is emerging as preferred data format for storing information on RFID tags. RFID and XML can work together as complimentary technologies. The idea is central to our proposed approach and is introduced in this research as a new opportunity for providing RBAC with multi-layer encryption of the data stored in RFID tags, without any cryptographic computations on the tag. Our proposed approach is based on this thorough review of related technologies and standards.

Chapter 3

Proposed System Model and Framework

3.1 Introduction

In this chapter, we motivate our design by presenting a scenario that RFID tags are employed as distributed storages. The following example is based on an on-going project at Concordia University that aims at providing real-time services to facilities managers and fire fighters. These two roles will be used as examples of roles used in RBAC in Section 4.1

During fire emergencies, it is critical for fire fighters to accurately identify their positions in the building in real-time and obtain detailed and up-to-date information about the circumstances. The availability of such information directly affects the correctness of the decision that fire fighters make and the effectiveness of operations that they undertake. Currently, upon arrival in the fire site, the fire fighter are given a set of floor plans of the building, including special symbols about the location of chemical, hazard material, etc. There are a several problems in this solution. First, a fire fighter has to identify the position of the floor plan, move to that place, and then find the fire source. However, it may not be easy to complete all these tasks efficiently during a fire emergency due to time pressure. Besides, these floor plans may include certain confidential information (e.g., hazard

materials) which adds additional security concerns with handling these documents. An alternative solution is to maintain the floor plans of all the buildings in the city at the fire station. Hence, whenever the fire station receives a fire alarm from a specific building, the corresponding floor plan is loaded to fire fighters' portable devices (e.g., a PDA) before departure. This solution does not provide real-time positioning within a building. Another issue is the overhead of keeping all the floor plans maintained at the fire station up-to-date. Most of the information about the building that would be useful to fire fighters would also be useful to facilities mangers except some sensitive information such as hazard materials that would be useful only to specific inspectors for health and safety department.

To address the weaknesses of current solutions, we use RFID tags as the media storing the floor plans and other information. Because of the convenience of wireless communications, a fire fighter or facility manager can readily download the necessary information and display it on the portable device when the tag is within a certain range. Figure 3.1 shows a sample structure of a Building Management (BM) document written in XML. Through accessing information stored under certain tags, e.g., $\langle Map \rangle$ and $\langle Hazard - Material \rangle$, a floor plan can be generated. There has been research work and ongoing activities in standardizing the representation of floor plans in a specific format, e.g., XML [72]. To make navigation possible within the building, CAD floor plans are needed on mobile devices to display the location of user and to generate path to other positions within the building. This information should be in appropriate format in order to calculate routes and data analysis. Floor plans in an image format such as JPEG or GIF do not work adequately. Standardized data formats are preferred, but several requirements are not yet fully supported by available standards [66]. Detailed discussions about this issue is out of the scope of this thesis. Moreover, upon receiving the signals from multiple tags, the fire fighter can identify his/her current position and the path to point of interest. A brief introduction about INS was discussed in Section 2.1.3.

We notice that there is still a problem remained in the above solution, i.e., providing part of the data on the RFID tags to a certain group of users (e.g., fire fighters, facility managers, Health and Safety (H&S) inspector or regular user of the building) while at the same time preserving the confidentiality of certain private information that would be available only to a specific group of users.

```
<Building-Management Name="EV Building">
    <Environment-Data>
        <Map>
            <Nodes>
                <Node Number= "">
                    <x></x>
                    <y></y>
                </Node>
            </Nodes>
            <Links>
                <Link Number="">
                    <Start-Node></Start-Node>
                    <End-Node></End-Node>
                </Link>
            </Links>
        </Map >
        <Floor Number="" >
            <Department></Department>
            <Room Number=""></Room>
        </Floor>
    </Environment-Data>
    <Inspection>
        <Hazard-Material>
            <Room Number=""></Room>
            <Type></Type>
        </Hazard-Material >
        <Product>
            <ID></ID>
            <Product-Specification></Product-Specification>
            <Status></Status>
        </Product>
        <History>
            <Comments></Comments>
            <Date></Date>
            <Inspector></Inspector>
        </History>
    </Inspection>
</Building-Management>
```

Figure 3.1: XML Document about Building Management

3.2 System Design

XML has emerged as the de-facto standard for storing and exchanging information in vast number of applications, including the new type of applications that use RFID tags as distributed storage [27, 46]. In this thesis, we focus on RFID-based data stored in the XML format. However, the same framework and data encryption scheme can be readily applied to other data formats that support hierarchical representation. Moreover, we assume that the structure of the XML document and the corresponding XACP are changed infrequently. Further, we assume that the XACP is defined in terms of RBAC [20].

We assume that there exists a trusted data provider, who is responsible for providing various services using RFID-based distributed storage, such as:

- *Defining ACP*: This step includes defining permissions for users to access various parts of data stored on an RFID tag for different operations.

- *Managing Roles*: This service defines roles and associates users with roles. Example of different roles along with there access permissions is shown in Table 3.1.

- *Data Management*: This process develops data architectures, practices and procedures dealing with data and then executing these aspects on a regular basis. The type, amount and order of information to be stored on the RFID tag is also defined in this process.

- *Organizing RFID tag*: As per the requirement of our application, we use different types of tags with varying memory sizes. The type of tag, its usage and corresponding location within the building is decided in this process. For example, in Figure 3.2 active tags with large memory are located at the entry points of the floor; one active tag with small memory is assigned per room to store required information about the room and its user; and passive tags are attached at doors to assist in tracking.

Table 3.1: Examples of Different Roles and access permissions

Role Name	Description of access permissions
Fire Fighter	Access to all floor plans, building information and sensitive information like chemicals in various laboratories
General User of the Building	Access to general floor plans only
FM Inspector	Access to general floor plans and building specific information

As the main focus of this research is on securing the RFID data, we assume that an XACP has been defined to control the access to different parts of data stored on the RFID tags. Thus, the goal of our work is to derive a data protection scheme that can strictly enforce this XACP and at the same time minimize the storage.

We assume that an adversary can be an outsider or a user who is assigned with one or a specific set of roles according to the XACP. We also assume that the adversary has certain capabilities of launching a physical attack [2, 24, 83] so as to obtain the content stored on an RFID tag, even if the tag is protected by an authentication scheme.

We assume the existence of a semi-trusted server, which will execute assigned functionalities honestly, but it is not trusted to access confidential data stored on RFID tags. The main responsibility of this server is key management, the server stores public information associated with keys. Key generation is discussed in Subsection 3.3.4

3.2.1 Overview of the Proposed Framework

Figure 3.3 shows the proposed framework for protecting RFID-based distributed data. The framework can be divided into three parts: First part takes care of all the processes required before storing data on the RFID tags, such as generating a set of role keys based on the structure of the XML document to be protected and the corresponding XACP, and then applying multi-layer encryption with these keys. Second part is about releasing the encrypted XML document on RFID tags, which can be downloaded by any RFID reader. Third part of the framework takes care of decrypting the XML document. However, the holder of RFID

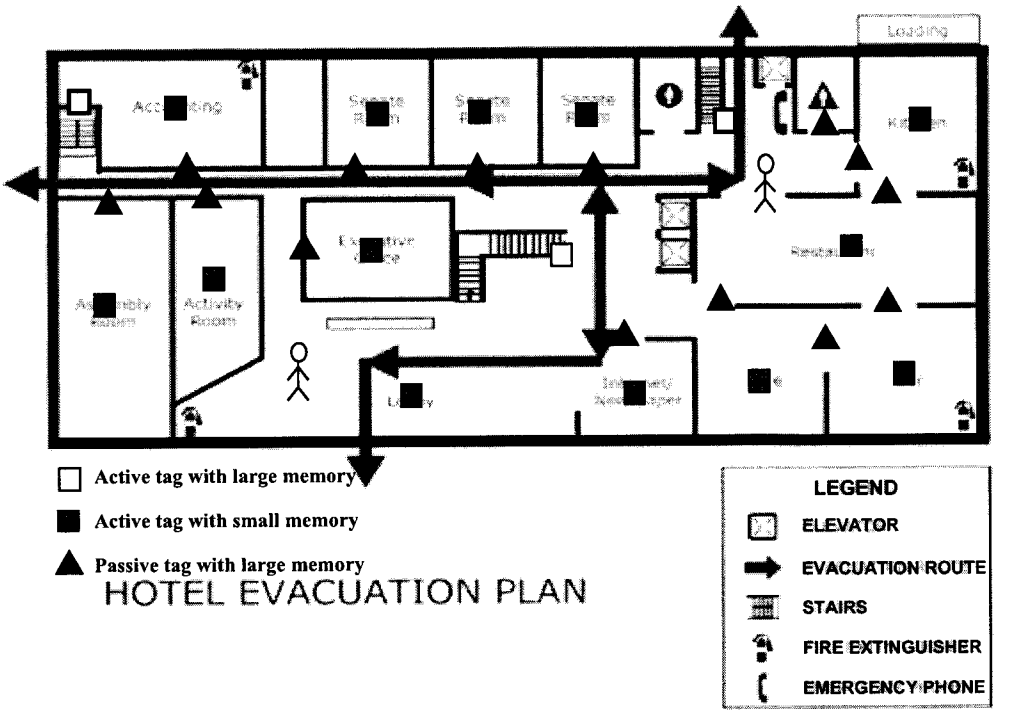


Figure 3.2: Floor Plan Example Using Different Types of RFID Tags

reader can only partially decrypt the encrypted document with regard to the roles that have been assigned to him/her. In case of complex policies, the structure of the original XML document changes. Complex policies and structure changes are discussed in more detail in Sections 3.3.2 and 3.3.3, respectively. The re-generation of original structure of XML document is also covered in the framework.

The core design of this framework lies in the tasks performed by the data provider. Once it is completed, the tasks of releasing, downloading and decrypting are straightforward in simple policies. The decryption requires some computation on the client side in case of complex policies. Thus, in the following sections, we focus our discussion on data encryption and decryption.

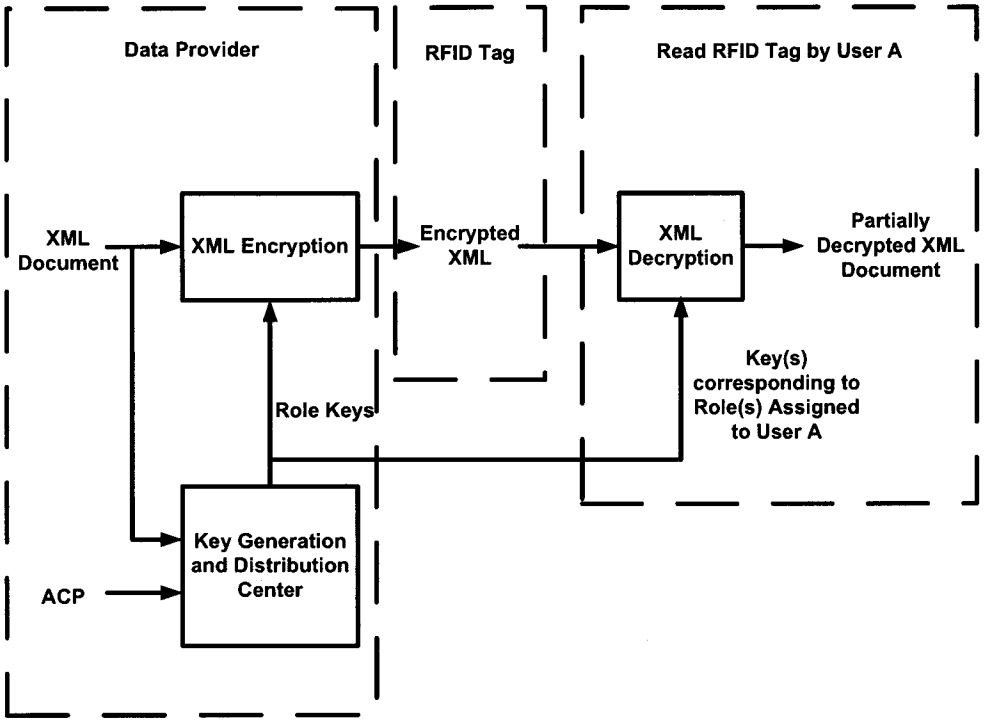


Figure 3.3: XML Protection Framework

3.3 Data Encryption Procedure

As shown in Figure 3.4, the procedure of data encryption consists of the following steps: *access tree generation*, *policy type checking*, *access tree transformation*, *key generation*, and *XML encryption*. Among them, the access tree transformation step is applied only when the XACP is identified as a complex XACP through the policy type checking step.

3.3.1 Access Tree Generation

The inputs of this step include the original XML document, which is expected to be protected while providing services to the authorized users, and the corresponding XACP. One purpose of access tree generation is to simplify the tree structure of the XML document

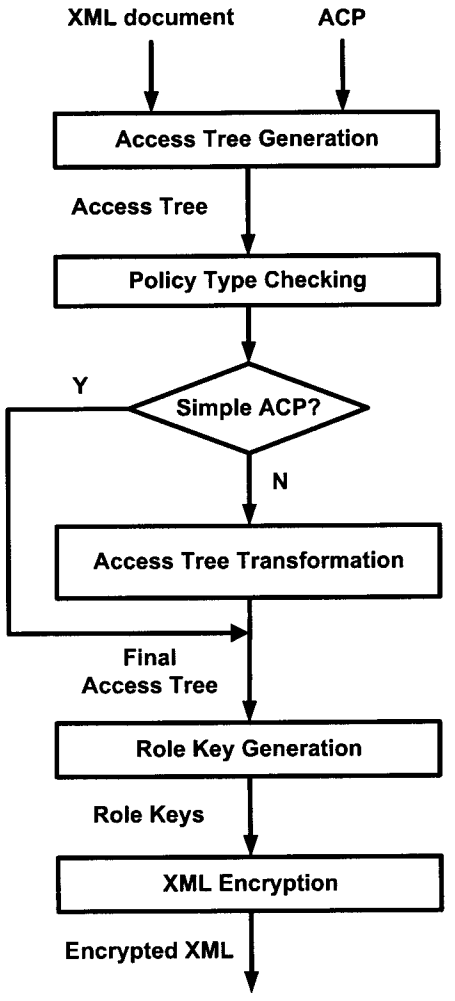


Figure 3.4: Process of XML Encryption

so as to reduce the complexity of the following task that determines the XACP type. Another reason is to reduce the number of encryptions so as to improve the storage efficiency. The basic idea is to compress a connected part of the original XML tree structure that are accessible by the same set of roles into a single node. The output is called an *Access Tree*.

Due to its hierarchical nature, an XML document can be represented as non-cyclic directed tree $T = (N, B)$ where N is the set of all the nodes $N = \{n_1, \ldots, n_n\}$ in T and B is the set of all the branches $B = \{b_1, \ldots, b_m\}$ in T. In our context, the XACP is represented with a set of subject-object relations, where a subject is a role and an object is

a set of elements within the XML document that are accessible by this role. In the context of $T = (N,\ B)$, we define the following notations:

- $Prnt(n_i,\ T)$ is defined as the parent node of n_i in T.
- $Chld(n_i, T)$ is defined as the set of all the children nodes of n_i in T.
- T_{n_i} is defined as a subtree of T, the root node of which is n_i.
- n_{root} denotes the root node of T.

For simplicity we will use $Prnt(n_i)$ and $Chld(n_i)$ for node n_i in T, instead of $Prnt(n_i,\ T)$ and $Chld(n_i, T)$, in the remaining part of this thesis.

Let $R = \{R_1, \ldots,\ R_r\}$ denote the set of all the roles defined in the XACP. Let $\{R_i,\ N_i\}$ denote a subject-object relation, where R_i is a specific role in R and $N_i = \{n_i^1, \ldots,\ n_i^k\}$ is the set of nodes in T that are accessible by users assigned with role R_i.

Taking T as the input, Algorithm 1 is executed to mark each node of T (denoted as n_i) with a set containing the roles that can access this node, which is called as the *Role Set* of node n_i and is denoted as $M(n_i)$.

Algorithm 1 Role Marking

Require: The tree representation of an XML document $T = (N,\ B)$, the XACP of T that is represented with a set of $\{R_i,\ N_i\}$ pairs.
Ensure: The XML document T, each node of which has been marked with roles that can access it.

```
1: for all n_i in N such that 1 ≤ i ≤ n do
2:    Initialize M(n_i) as a empty set.
3: end for
4: for all R_i in R such that 1 ≤ i ≤ r do
5:    for all n_i^j in N_i such that 1 ≤ j ≤ k do
6:       Add role R_i to M(n_i)
7:    end for
8: end for
```

After all the nodes are marked, Algorithm 2 is performed from the root of T, i.e., $ATG(T,\ n_{root})$. At each step of the search, the marking set of the current node n_i, i.e., M_i,

is compared to that of the next node n_j, i.e., M_j. If $M_i = M_j$, the branch between n_i and n_j is removed, and n_j is merged into n_i. At the end of this depth-first-search, the access tree corresponding to T is generated.

Algorithm 2 Access Tree Generation $ATG(T,\ n_G)$

Require: The tree representation of a XML document $T = (N,\ B)$ in which each node is marked with the set of accessible roles
Ensure: The access tree T^A corresponding to T

```
1: for all n_i in Chld(n_G) do
2:   if M(n_i) = M(n_G) then
3:     for all n_j in Chld(n_i) do
4:       Prnt(n_j) ← n_G
5:       Add n_j to Chld(n_G)
6:     end for
7:   else
8:     Call ATG(T, n_i)
9:   end if
10: end for
```

For example, the tree structure of an XML document D is shown in Figure 3.5. Given that, there are four roles involved in the XACP of D, denoted as R_1, R_2, R_3, and R_4. The detailed access information about each role is shown in Table 3.2.

Table 3.2: Role-based Simple XACP

	Accessible Nodes
R_1	$A,\ B,\ D,\ E,\ I,\ J,\ K,\ L$
R_2	$A,\ C,\ F,\ G,\ H$
R_3	$A,\ B,\ C,\ D,\ E,\ G,\ I,\ J,\ L$
R_4	$A,\ B,\ C,\ D,\ E,\ F,\ G,\ H,\ I,\ J,\ K,\ L$

The step of access tree generation is shown in Figure 3.6.

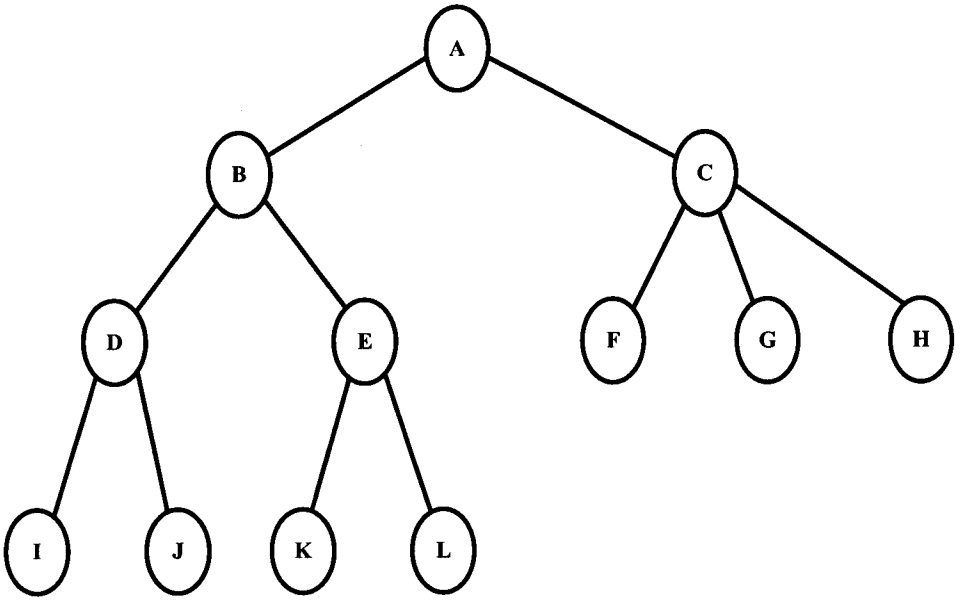


Figure 3.5: Tree Structure of XML Document

3.3.2 Policy Type Checking

Previous work on XML access control [10, 51] focus on scenarios where data access can be defined with a simple XACP. As to complex XACPs, which happen frequently in real-world applications, Crampton briefly described two approaches [10]. One is from the role-based aspect, while the other is cryptography-based. Instead of encrypting data, the former generates different views of the same document for different roles. It is fine with the applications of data outsourcing over the Internet [10, 51], where storage is not a big concern, but is unsuitable for the new RFID-based data storage applications [27, 46]. As to the latter, as indicated by Crampton, the usage of cryptography in handling a complex XACP is problematic [10]. Moreover, another important issue is that previous definitions of simple XACPs and complex XACPs are inaccurate, and thus fail to draw a clear line between

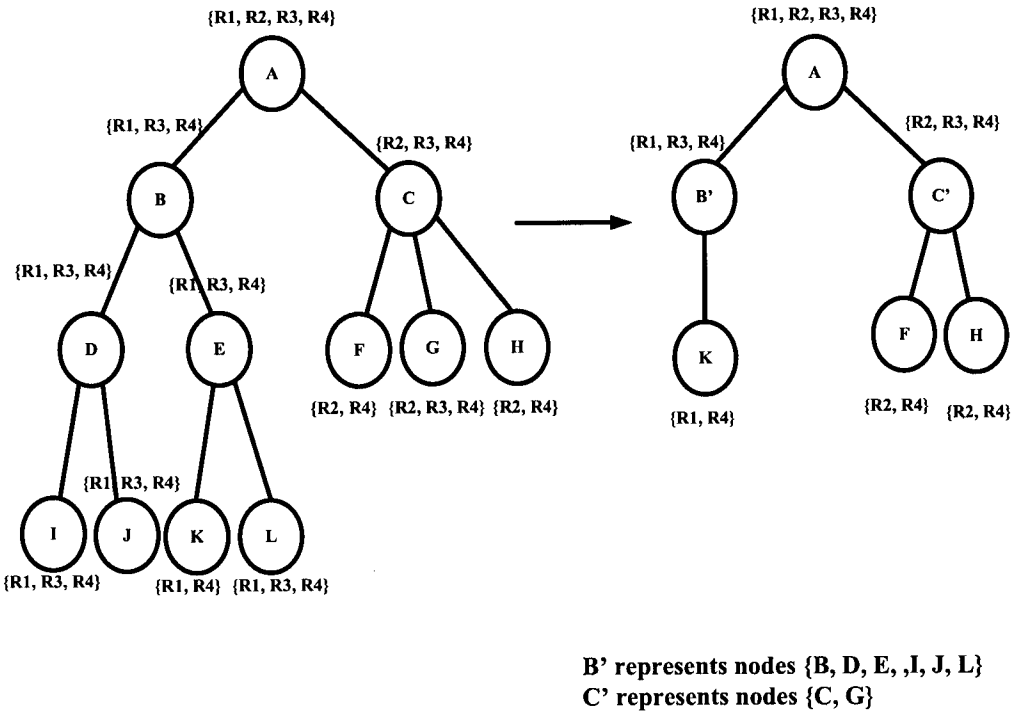


Figure 3.6: Access Tree Generation (Simple XACP)

them. Therefore, we redefine these two terms as follows.

Definition 1 *The ACP of an XML document D is called a* Simple XACP, *if the role set of any node in the access tree of D is a subset of that of its parent node, if any. Otherwise, it is called as a* Complex XACP.

Given an access tree, Algorithm 3 is executed to determine the exact type of the associated XACP, simple or complex. Let N^A be the set of all the nodes in the acces tree T^A. We continue with the example in Section 3.3.1 with a small change, namely, adding a new role R_5 which can access node A, C, D. Figure 3.7 shows the process of access tree generation for this new XACP. In Figure 3.7, the role set of node D is not a subset of that of its parent node, i.e., node B. Thus, this new XACP is complex. The policy checking algorithm is given in Algorithm 3.

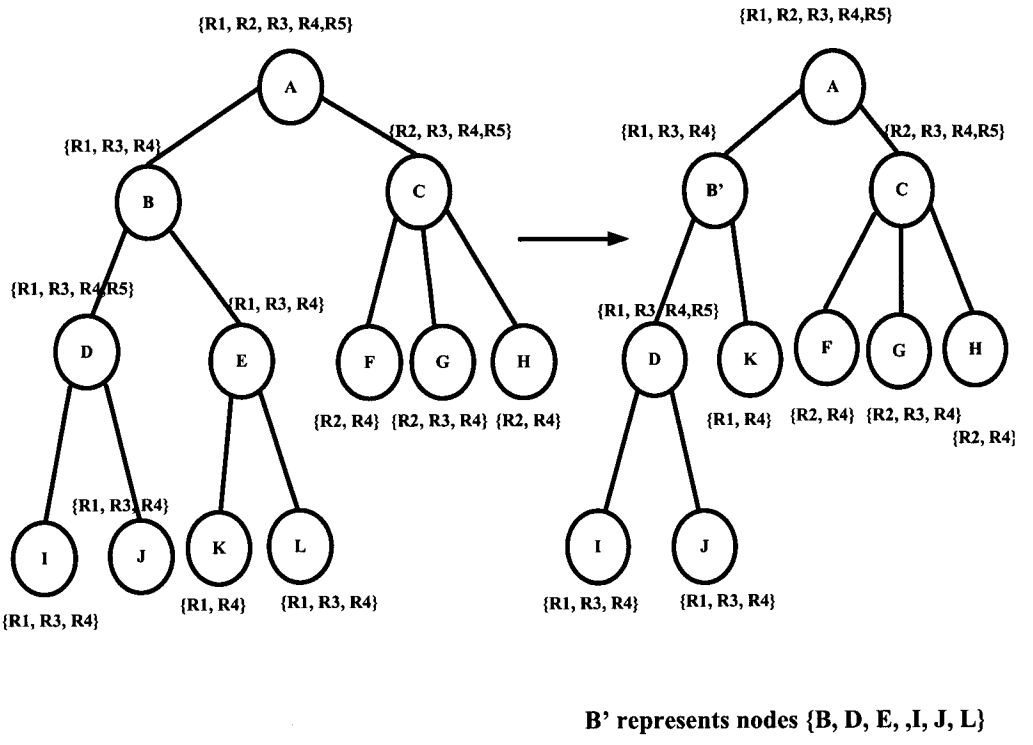


Figure 3.7: Access Tree Generation (Complex XACP)

3.3.3 Access Tree Transformation

We observe that, in general tree structure encryption, to encrypt a node literally means to encrypt only the content of that particular node. Contrarily, in XML encryption [79], to encrypt a node n_i means to encrypt both n_i and all its child nodes, i.e., T_{n_i}.

Given that an XML document is protected by a complex XACP, there exists at least one role that can access a specific node in the document, say A, but fails to access its parent node. As a result, it is impossible to perform multi-layer encryption according to the hierarchical structure of the access tree, or say the XML document, since to access the content at an inner layer requires keys of both inner and outer layers. However, it does not cause any problem if the XACP is simple. For this reason, the step of access tree transformation is optional, and will be executed only when the XACP is complex.

Algorithm 3 Policy Type Checking

Require: The access tree of a XML document T^A.
Ensure: Return the type of XACP denoted as *Type*.

```
1: Type ← Simple
2: for all n_i in N^A do
3:   if n_i is not the root node of T^A then
4:     if M(n_i) ⊈ M(Prnt(n_i)) then
5:       Type ← Complex
6:     end if
7:   end if
8: end for
9: return Type
```

To address this issue, the original XML document has to be transformed in such a way that a simple XACP is enforced on the transformed document, which is equivalent to the complex XACP enforced on the original document. The transformation can be done through either copying or moving certain parts of the document to other positions. In RFID-based distributed storage applications, we are more concerned about storage efficiency. Hence, our transformation algorithm, i.e., Algorithm 4, involves only moving operations. More specifically, Algorithm 4 is executed from the root node of the access tree (denoted as n_A), i.e., $ATT(T^A,\ n_A)$, to complete the task of transformation.

In terms of the complex XACP example described in Section 3.3.2, the resulting transformed access tree is shown in Figure 3.8.

3.3.4 Role Key Generation

To reduce the key management overhead, it is desirable that each role maintains a minimum number of keys. Based on the scheme proposed by Akl and Taylor [1], Crampton presented a method that applies hierarchical and RBAC to XML documents in terms of simple XACP that focuses on Internet data outsourcing applications [10]. In our framework, through identifying complex policies and making appropriate transformations, Crampton's scheme can be applied to the resulting access tree, no matter which type of ACP needs to

Algorithm 4 Access Tree Transformation $ATT(T^A,\ n_T)$

Require: The access tree T^A corresponding to T
Ensure: The transformed access tree T^T

```
1: for all n_i in Chld(n_T) do
2:    Call ATT(T^A, n_i)
3: end for
4: if n_T is not the root node of T^A then
5:    if M(n_T) ⊈ M(Prnt(n_T)) then
6:       n_M ← Prnt(n_T)
7:       Done ← false
8:       while n_M is not the root node of T^A and Done = false do
9:          n_N ← Prnt(n_M)
10:         if M(n_T) ⊆ M(n_N) then
11:            Prnt(n_T) ← n_N
12:            Add n_T to Chld(n_N)
13:            Done ← true
14:         else
15:            n_M ← n_N
16:         end if
17:      end while
18:   end if
19: end if
```

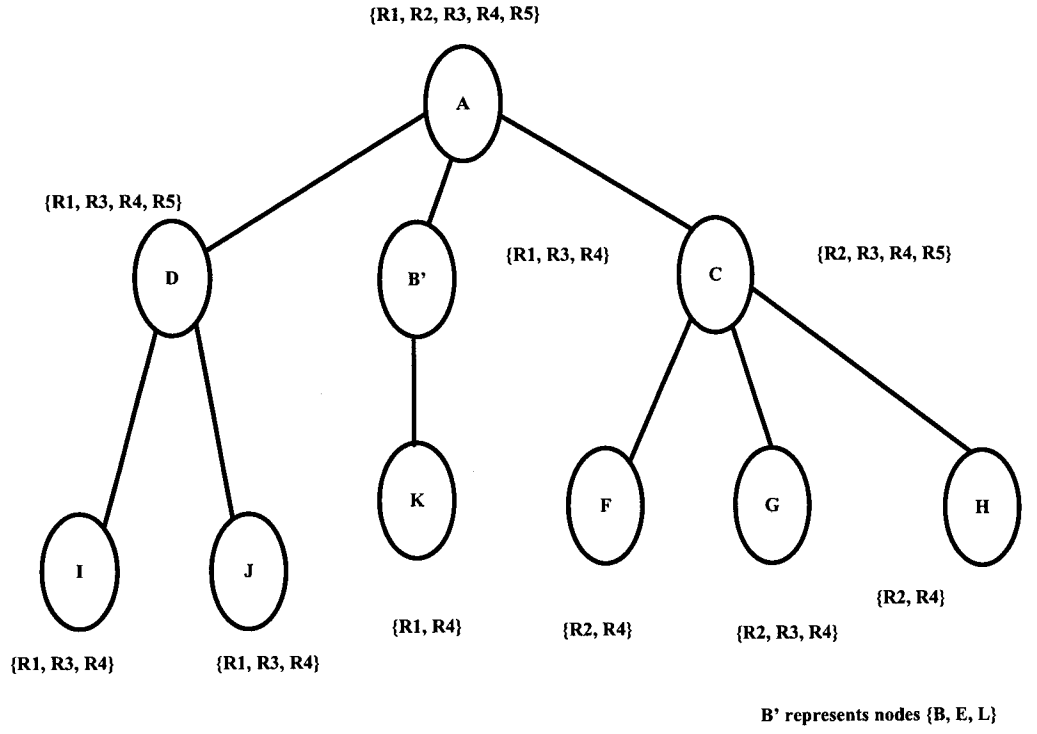
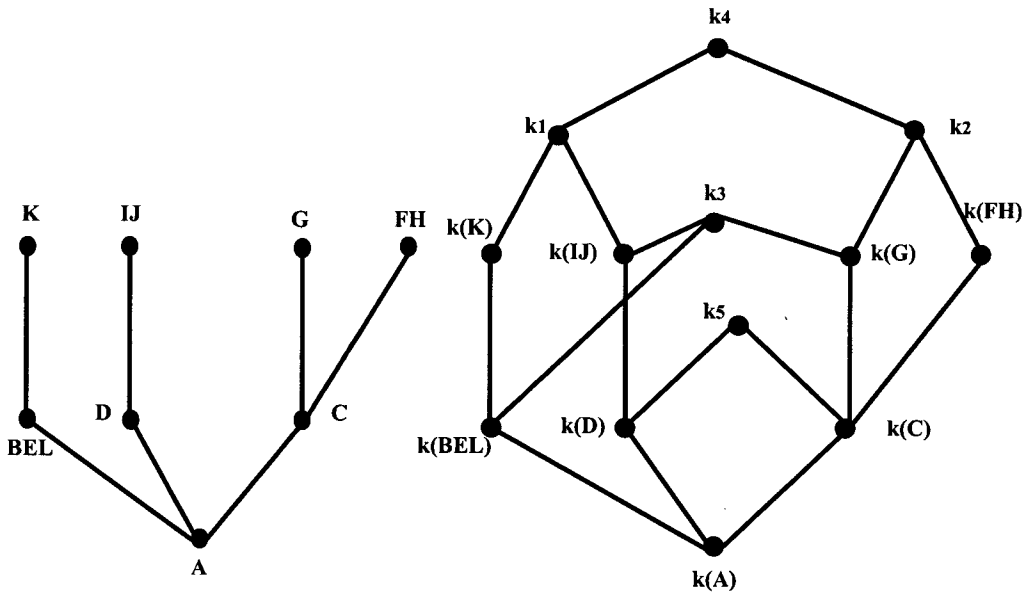


Figure 3.8: The Process of Access Tree Transformation

be enforced on the original XML document.

Take the resulting access tree in Figure 3.8 as an example. After further simplification through combining leaf nodes that share the same parent node and have the same role set, the resulting policy hierarchy and key hierarchy corresponding to T^T are shown in Figure 3.9. Each role R_i needs to maintain only one key, i.e., k_i. Different parts of the document that are accessible by users with Role R_i can be decrypted by either k_i or other keys derived from k_i. Note that, keys in the key hierarchy are arranged based on the partially ordered set so that the key at a higher level, say n_i, can be used to derive any key at lower levels, say n_j, if there is a directed path from n_i to n_j. For example, a user with Role R_5 is assigned with k_5, from which the user can derive $k(A)$, $k(C)$, $k(D)$.

Most key assignment schemes seek to minimize the number of keys that need to be



(a) Policy Hierarchy **(b) Key Hierarchy**

Figure 3.9: Role Key Generation

distributed to users. This entails either making certain additional information public or providing each user with additional secret information (or both). In general, a key assignment scheme (or scheme) for an information flow policy $(L, \leq)$ defines four algorithms [11]:

- makeKeys returns a labeled set of encryption keys $(\kappa(x) : x \in L$, denoted as $\kappa(L)$;
- makeSecrets returns a labeled set of secret values $(\sigma(x) : x \in L$, denoted by $\sigma(L)$;
- makePublicData returns some set of data Pub that is made public by the trusted centre;
- getKey takes $x, y \in L, \sigma(x)$ and the public data, and returns $\kappa(y)$ whenever $y \leq x$.

To better understand the full process of role key generation we now describe Akl and Tylor scheme, as illustrated by Crampton with an example. Let χ denote the key hierarchy.

The data provider performs the following set of steps before generating the keys:

1. Choose large primes p and q and publish $n = pq$
2. Choose $\kappa \in [2, n-1]$ such that $(\kappa, n) = 1$
3. For each $x \in \chi$, choose a distinct prime $p(x)$
4. For each $x \in \chi$, define and publish $\pi(x) = \prod_{y \not\leq x} P(y)$
5. For each $x \in \chi$, compute secret key $k(x) = \kappa^{\pi(x)} mod n$

In this scheme one secret and one public value are assigned with each key. The secret value κ is known only to users and public value π is published by *semi trusted server.* Figures 3.10 and 3.11 show how values of p and π are associated with each element of the hierarchy. Given key $k(x)$, it is possible to derive key $k(y)$, where $k(y) \leq k(x)$, by computing $k(x)^{\pi(x)/\pi(y)} = (\kappa^{\pi(x)})^{\pi(x)/\pi(y)}) = \kappa^{\pi}(y) = k(y)$. Hence $k4$ is defined to be κ, $k1$ is $\kappa^{2.19.23.29.31.37.41}$, and so on.

The $\pi(y)$ value is public information and $\pi(y)$ is divisible by $\pi(x)$ whenever $k(y) \leq k(x)$ by construction. Note that it is not feasible to derive a key $k(z)$, where $k(z) \geq k(x)$, because this would entail computing integral roots of $\kappa mod n$. This method is also secure against a set of users pooling information in an attempt to derive keys for which they are not authorized [10].

3.3.5 Multi-Layer Encryption

Given a transformed XML tree T^T and role keys, we now illustrate how to generate an encrypted XML document in accordance with the given XACP such that a user has access to only those parts of document defined in XACP, provided she/he knows the corresponding role key.

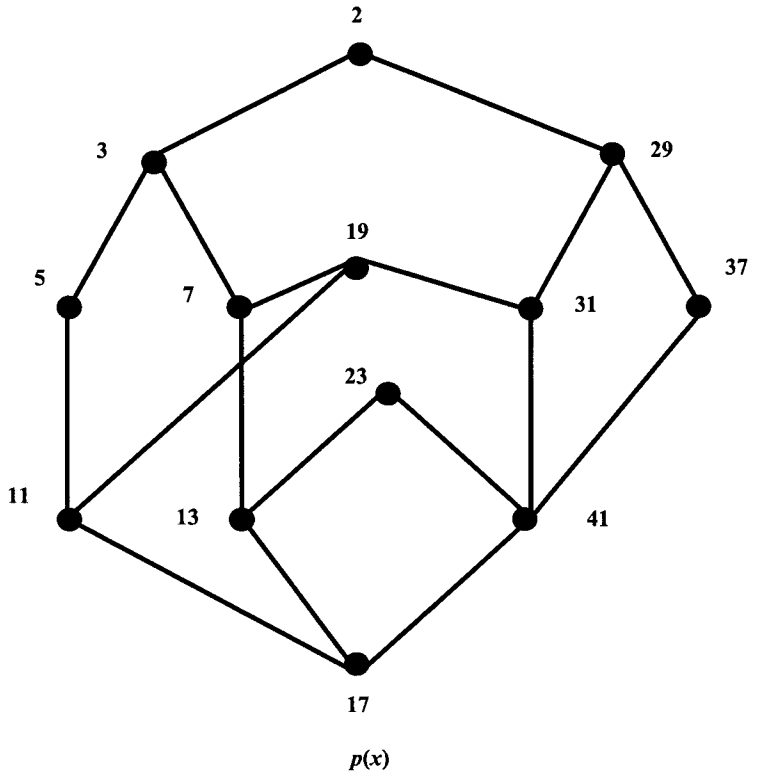


Figure 3.10: Assigning Prime Value to Each Node (Using Akl and Tylor Method [1])

W3C defines a standardized schema for representing encrypted data, cryptographic keys and encryption algorithms in XML form, as part of recommendation on XML Encryption syntax and processing. EncryptedType is the abstract type from which EncryptedData is derived. EncryptedData is a basic component, containing four sub-elements:

- *EncryptionMethod*: is an optional element that describes the encryption algorithm applied to the cipher data. If the element is absent, the encryption algorithm must be known by the recipient or the decryption will fail. In our experiments it is always AES with 128-bit keys

- *KeyInfo*: is an optional element that carries information about the key used to encrypt the data. Subsequent sections of this specification define new elements that may appear as children of KeyInfo. In our framework, we use symmetric encryption scheme

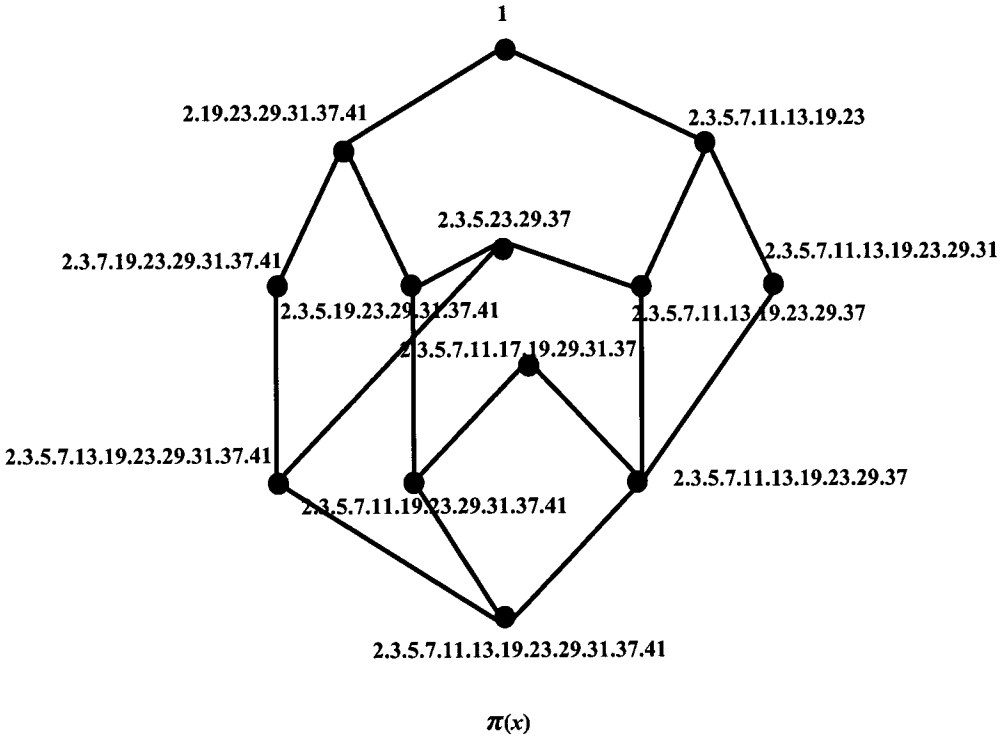


Figure 3.11: Assigning π Value to Each Node (Using Akl and Tylor Method [1])

therefore KeyInfo element is not explicitly defined in the encrypted document

- *CipherData*: is a mandatory element that contains the CipherValue or CipherReference with the encrypted data

- *EncryptionProperties*: can contain additional information concerning the generation of the EncryptedType (e.g., date/time stamp)

The actual XML multi-layer encryption is a recursive process of encrypting a same node more than once, i.e., super encryption as explained in Section 2.2.1. One of the biggest limitations of super encryption is that it increases the size of XML document exponentially. In order to reduce the number of encryptions we merged those nodes with their child nodes which are accessed by same set of roles, as described in Section 3.3.1. Notice that after performing Algorithm 2 all the merged nodes are considered as single node. The

encryption proceeds as follows: All nodes part of the same role set are encrypted with the same key. A node is being encrypted every time its parent node gets encrypted. The number of encryption layers of a node apart from its own encryption is equal to the number of its parent nodes.

For example, Figure 3.7 shows a tree generated from XML documents after performing Algorithm 2. Node G is a leaf node and after encrypting with key say $k(G)$, it will be encrypted by keys $k(C)$ and $k(A)$ because of nodes C and A. In order to decrypt node G, keys $k(A)$, $k(C)$, $k(G)$ are required. Firstly $k(A)$ is used, once decrypted, node A and its attributes are revealed. G's contents however are still encrypted: both child of node A are EncryptedData elements, while applying keys $k(C)$ and $k(G)$ respectively, contents of node G gets revealed.

3.4 Decryption and Regenerating Tree Structure

After conducting experiments with the proposed framework, we came across an issue after decryption, related to structure of XML document on the user side in complex XACP. In order to enforce a complex policy we first convert it into a simple policy by reforming the original structure of XML document. Because of these changes, it may be difficult for users to draw parallels between information stored in different parts of the XML document. For the example in Figure 3.7, in the original XML structure, node D is a child of node B and in general child node contains information related to its immediate parent node. But After access tree transformation to accommodate a complex policy, node D becomes child of node A as shown in Figure 3.8, situations like this makes it difficult to understand the hierarchy of information. In the following paragraphs we discuss XML decryption standard recommended by W3C [79] and propose a solution for regenerating the original XML structure after decryption.

XML Decryption Specification [79]

For each EncryptedData element to be decrypted, the decryptor must:

1. Process the element to determine the algorithm, parameters and KeyInfo element to be used. If some information is omitted, the application must supply it.

2. Locate the data encryption key according to the KeyInfo element. If the data encryption key is encrypted, locate the corresponding key to decrypt it.

3. Decrypt the data contained in the CipherData element.

    - If a CipherValue child element is present, then the associated text value is retrieved and base64 decoded so as to obtain the encrypted octet sequence.
    - If a CipherReference child element is present, the URI and transforms (if any) are used to retrieve the encrypted octet sequence
    - The encrypted octet sequence is decrypted using the algorithm/parameters and key value already determined from steps 1 and 2

Regenerating Original XML Structure

The XML decryption process itself is very straightforward. Consider the transformed access tree in Figure 3.8, after decryption, role R_5 is able to access nodes A, C and D. In this case, everything works fine and according to the XACP as role R_5 is authorized for only those decrypted nodes. As R_5 is not permitted to access node B, therefore its original position in the XML document and its relation with node D is of no concern to this role. Most of the XACP are a mixture of simple and complex, the roles which are governed by simple polices require the exact pattern in which the original information was stored. Consider the same example, after decrypting, role R_3 generates nodes A, B, C, D, I, J, G but it is not possible to know the original position of node B for R_3.

One of the simplest solutions would be to store two documents one for each simple and complex policies, this works exactly like storing different views of the same document in web-services. In our application, we have the constrain of the limited memory on RFID tags. To get rid of this limitation, we propose to add a meta node with every node that is displaced from its actual position. The meta node contains information that identifies actual position and helps in regeneration of original XML structure.

Now we describe this process with an example, before storing information in the meta node, we perform the following steps:

- Starting from the root node, assign a level to every node such that root node is at level one and for all the other nodes set the level one greater than the level of its parent node.

- Starting from the root node, for all nodes with same parent at same level, assign a number such that leftmost node is zero and each subsequent node on the right has a number one greater than the number of its left side node. Then start with each child node as root and assign number to its child. We call these numbers position numbers.

Figure 3.12 shows levels and position numbers assigned to Figure 3.7, before transformation. Going back to the same example, with levels and position numbers assigned, the following information is required to put node D in its original state:

- Level and position number of node B (the parent node of node D)

- Position number of node D

The level of parent node helps in adding a direct branch between a displaced node and its original parent node. The level of parent node of node D is 2, which identifies the level of node D is 3 in original structure. In general there is more than one node at certain level; position number of parent node is required to find out the exact position of the parent out

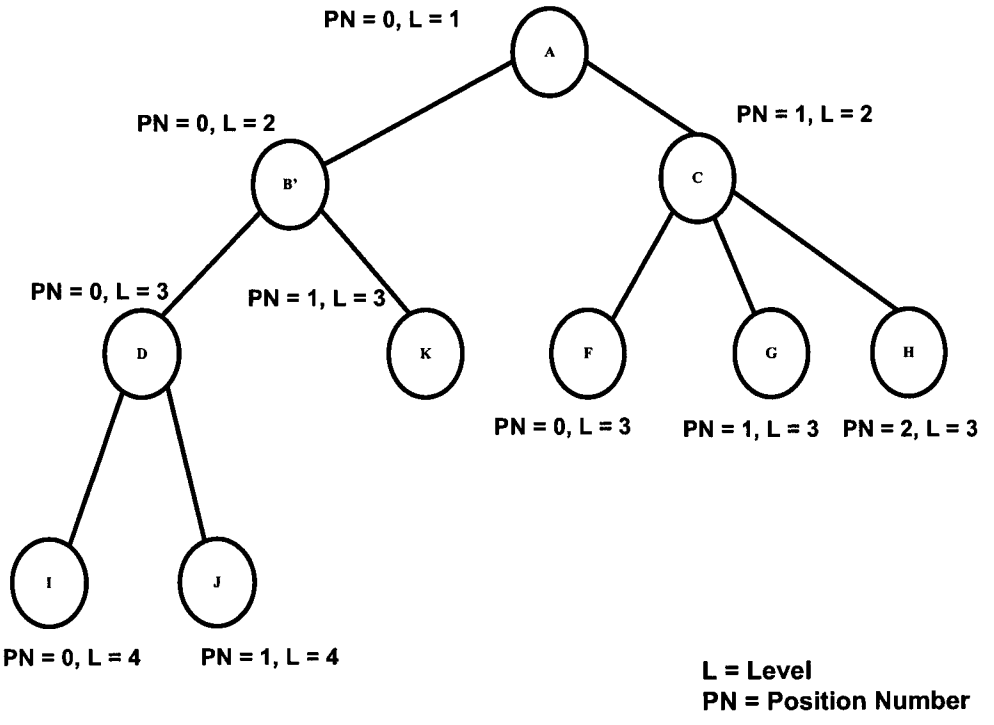


Figure 3.12: Level and Position Number Assignment

of many nodes at the same level. In the given example, there are two nodes at level 2 and position number of node B is zero. This implies that node B is the left most child of node A. The position number of a displaced node helps in locating its exact position in the original structure. The position number of node D is zero, i.e., it is the left most child of node B. With these three parameters the original XML structure can be regenerated.

This extra information can be stored as shown in Figure 3.13. A meta node does not need to be encrypted explicitly; it can be encrypted with the same key as its related displaced node. As it does not contain any confidential information, it can also be left unencrypted.

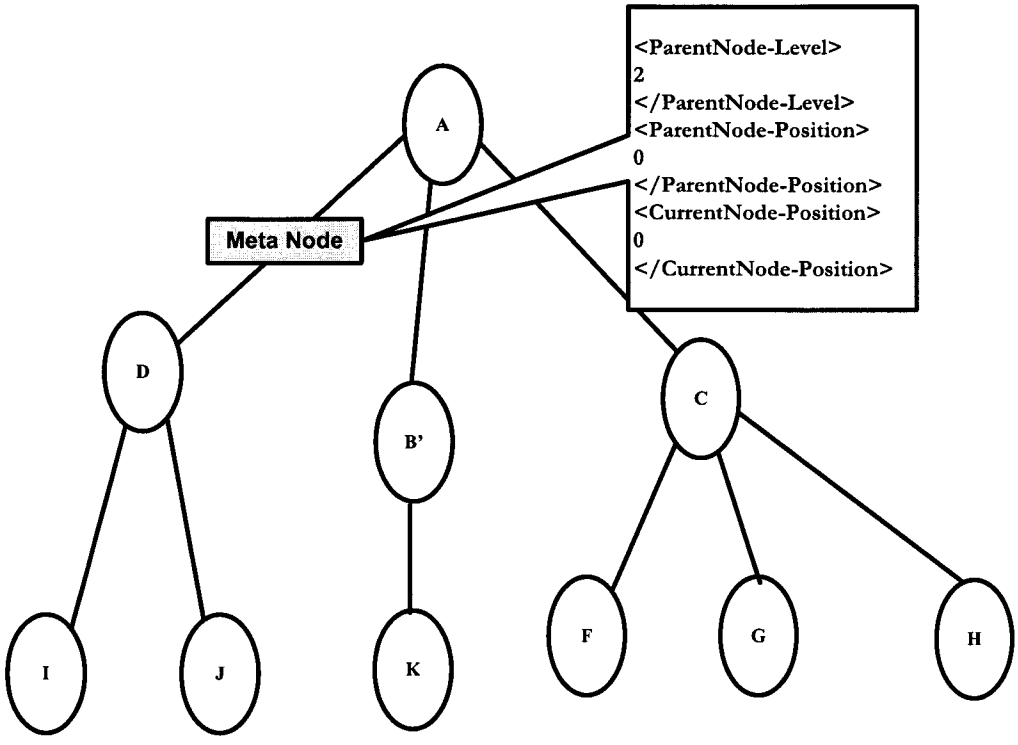


Figure 3.13: Transformed Access Tree with Meta-Node

3.5 Empirical Results

To evaluate the usability of the proposed access control framework and multi-level encryption scheme, we perform the following experiments, which are implemented in Java, to measure the storage efficiency of the proposed scheme.

The RFID tags used in our experiments are based on Intelligent Long Range (ILR) technology provided by IDENTEC SOLUTIONS. We use i-Q RFID tag which have range up to 100 m and have 32 KB memory. These tags are mounted on the walls of corridors and/or rooms in order to span the required space. We use an i-CardIII RFID reader, which can be embedded into a Computer or PDA processing unit via a PCMCIA (Personal Computer Memory Card International Association) slot.

Table 3.3: Encryption Modes

	XML Tag	Tag Content	Outermost-level Encryption Tag	Inner-level Encryption Tag(s) (if has more than one level encryption)	Ciphertext Encoding Conversion
Mode I	XML-Enc	XML-Enc	XML-Enc	XML-Enc	Yes
Mode II	Std-Enc	Std-Enc	Std-Enc	Std-Enc	No
Mode III	Std-Enc	Std-Enc	Std-Enc	No Encryption	No

As shown in Table 3.3, three types of encryption modes are implemented in our experiments. In Table 3.3, *XML-Enc* and *Std-Enc* denote XML Encryption [79] and standard encryption, respectively. For XML encryption, we use a Java implementation of Apache XML Project [21] that follows the W3C recommendation [79] and uses AES with 128-bit keys.

In our preliminary experiments, we encrypt an XML document with the structure shown in Figure 3.5 in two distinct ways so as to conform to the simple XACP shown in Table 3.2 and the complex XACP described in Section 3.3.2, respectively. In either case, we assign an equal amount of data to each node in the document, and then increase this amount gradually until the RFID tag memeory is full, i.e., 32 KB. Moreover, we assume that all the encryptions performed on this document use the same cryptographic algorithm (e.g., AES) and are with the same parameters (e.g., 128-bit key size).

In Figure 3.14 and Figure 3.15, we show the percentage of increase from the original XML document to the encrypted one under three encryption modes, given that the access control enforced by the encryption conforms to the required simple or complex XACPs, respectively. In both figures, the increase of the document size after encryption in Mode I is much higher than those of Mode II or Mode III. To better understand the results, in the next paragraphs each mode is illustrated separately along with its outcome.

Mode I

This mode is exactly the same as being defined and specified by W3C for XML encryption, also described briefly in Sections 3.3.5 and 2.2.1. Encrypting the content of an XML

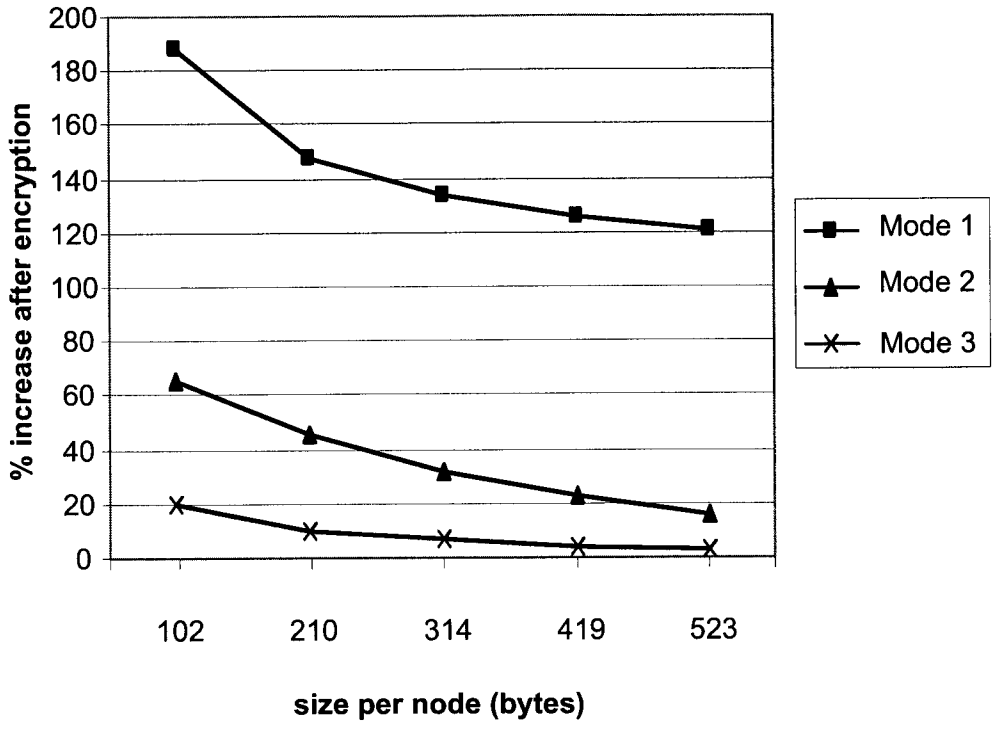


Figure 3.14: Storage Efficiency Under Simple XACP

document is performed as follows. Firstly the section of the document (the Element or Element content) to be encrypted are serialized (into UTF-8). This can be done using the canonicalization algorithms available for XML Signature. The resulting byte stream is then encrypted using the selected algorithm. The encrypted bytes are then base64 encoded and placed in a $< CipherValue >$ structure in an $< EncryptedData >$ element. If asymmetric encryption is used, a key is generated; this is in turn encrypted into an $< EncryptedKey >$ element which is added as a $< KeyInfo >$ of the $< EncryptedData >$. The resulting $< EncryptedData >$ replaces the nodes in the XML document that were encrypted [47].

The conversion from UFT-8 to base64 results in a rise up-to 33% of the original size in case of simple XML encryption. The problem further rises during multi-level encryption, since the increased size of the ciphertext becomes the cleartext for the next cycle of

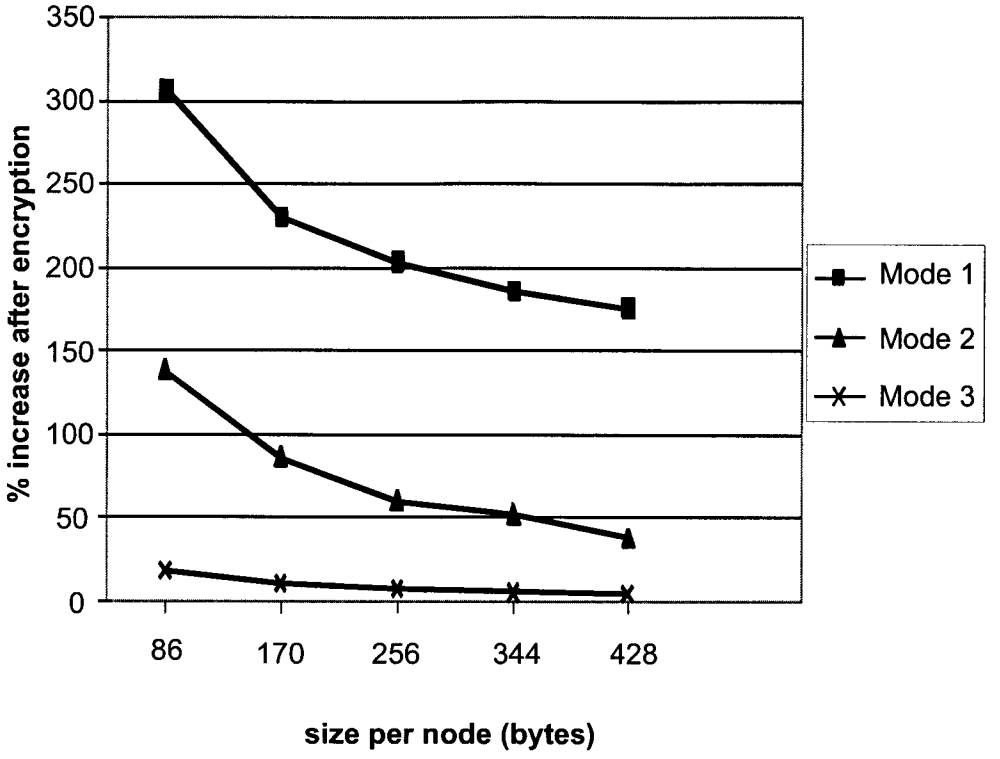


Figure 3.15: Storage Efficiency Under Complex XACP

encryption. In the process of super-encryption, the extra encryption tags added to represent the information required for decryption and to meet the specifications also get converted to base64 in each round of encryption, which results in extra overhead. For example, Figure 2.9 shows the encrypted value stored in $< CipherValue >$ after encrypting XML document shown in Figure 2.6, twice. After, first round of encryption there is an increase of 33%, this compounds the overall increase after second round.

Because of the above mentioned reasons, in Figures 3.14 and 3.15, the increases of the document size in Mode I are from 121% to 188% under the simple XACP and from 175% to 306% under the complex XACP, respectively.

Mode II

To overcome the limitation of encryption overhead in Mode I, we introduce a new method for encrypting. This new method makes use of all the predefined XML encryption standards, for example; elements like $< EncryptedData >$ to define type and $< Encrypted-Method >$ to represent the encryption algorithm used. The only difference between the two modes is conversion of encrypted text, i.e. Mode II does not conduct any conversion to base64 from UFT-8 after encryption.

The encryption is done by using a standard encryption procedure, the size of ciphered text in $< CipherData >$ would be slightly more than the original plain text. The extra encryption tags are treated the same way. The encrypted XML file looks exactly the same as shown in Figure 2.9, but there is a big difference in the size of both files. Figure 3.14 and Figure 3.15 show that the percentages of increase in Mode II are from 16% to 65% under the simple XACP and from 38% to 138% under the complex XACP, respectively.

Mode III

As mentioned earlier, storage space is one of the biggest concerns in our application. Though Mode II is able to reduce the encryption size to a very large percentage still we can further improve the storage efficiency. We noticed that, in both previous modes the extra encryption tags repeat the same information in every super-encryption cycle because we used symmetric encryption with same algorithm and key size in all the experiments. To get rid of this repeated information, we made some changes in Mode II.

In case of simple encryption, Mode II and Mode III work in the same fashion. Both use standard encryption procedure without conversion to base64 and add extra encryption tags around ciphertext. But during multi-layer encryption, instead of adding encryption tags with same information in every super-encryption round, we add only once after the last encryption round. The XML file looks exactly the same as it does in Mode I and II, but

the size is further smaller than in Mode II.

As a result, only the outermost-level encryption tag is retained. Figure 3.14 and Figure 3.15 show that the percentages of increase in Mode III are from 2.8% to 20% under the simple XACP and from 4.2% to 18% under the complex XACP, respectively. According to our experiments, Mode III is the best option for encrypting the XML document.

3.6 Challenges

Although the proposed framework can be implemented using existing RFID technology, there are many hindrances that need to be addressed to make it fully functional and more robust. Further development in the following areas would lead to less expensive hardware solutions, industry-wide standards and low-cost supporting software applications. The challenges can be categorized under the following main topics: (1) challenges related to adopting RFID technology; (2) XML challenges; and (3) technology adoption and social challenges.

(1) Challenges related to adopting RFID technology

The Following are some of the obstacles that need to be overcome in order to adapt RFID technology:

- *RF challenges*: Attaching an RFID to certain material reduces it signal strength. It is directly related to the effects of materials such as liquids or metal on electromagnetic waves that interfere with the operation of the RFID system and shorten the readability range.

- *Interference*: Reading one tag at a time makes it accurate but very slow. Radio signals transmitted simultaneously by different tags cause interference and collision,

lowering the quality of transmission and increasing the error rate. This requires anti-collision procedure, which considerably slow the system. Further, simultaneously reading large number of tags makes it difficult to find if any tag failed to get read.

- *Standards*: The lack of completer global standard adds to companies resistance to wide adoption of RFID technology. There is no consistent UHF spectrum allocation and power regulations and certifications also vary in different countries. In addition, vendors are concerned with the high patent royalty which becomes an obstacle to the development of RFID systems.

- *Cost*: Currently the cost of manufacturing and customization of tags is high. In addition, RFID systems require infrastructure to interconnect all the stakeholders to be able to communicate electronically. This infrastructure requires tremendous amount of design and implementation efforts. On the other hand, the intangible benefits of implementing RFID systems make the cost-benefit and ROI (Return on Investment) analysis more complicated.

  Furthermore, barcode systems have been already implemented by many enterprises. RFID is still at developing stage; therefore, enterprises will keep two systems to operate. If RFID reform happens, then for a period of time RFID and barcodes must work side by side.

- *Physical Protection*: Since the tags are attached to components throughout the life-cycle, proper physical protection (e.g., against temperature and material effects) is needed. Further, protection from adversaries is also required; attacker can physically destroy the tag and launch denial of service attack.

- *Memory*: Tag memory is another big limiting issue. Although, RFID active tags have the liberty over other technologies like Barcode, to store information but it is not enough to store large amount of data.

- *Data transfer speed*: The RFID system must support high data transfer speed in order to be able to access all the information in short period of time. The low data communication rate would decrease the expected efficiency of the proposed framework.

- *Interoperability*: The wide implementation of RFID systems requires more standards to cover all types of tags and frequencies. Moreover, the need for multi-protocol tags and readers is evident for interoperability of different systems.

- *Power*: Limited lifetime of battery-assisted tags is a challenge that should to be addressed. Hence the choice between active and passive tags depends on the application requirements.

(2) XML challenges

- *XML Encryption*: XML encryption specification has been standardized by W3C, but no standard set of APIs for XML encryption is available yet. Several projects provide XML encryption APIs, none of them is final yet. Further, in XML encryption, clear text uses 8-bits per character (UTF-8), while the ciphertext is represented using 6-bits per character (base64). When the ciphertext encoding is converted from base64 to ASCII, there is a 33% increase of size. Even worse, such an increase is compounded in multi-level encryption, which results in a large percentage of increase just after a few levels of encryption.

- *XML representation of BIM*: XML has emerged as a standard format of information exchange. The efforts for developing BIM standards in XML format are in their early stage and the available standards and implementation of BIM-based systems are not complete and thorough.

Adopting BIM standards has its own challenges and obstacles; issues such as industry acceptance, interoperability between existing software platforms, change management from conventional methods to new BIM, qualified human resources, legal considerations and initial cost to change (hardware, software, training and implementation) have to be tackled for industry-wide implementation of BIM.

(3) Technology Adoption and Social Challenges

- Wide Implementation of such systems would bring resistance from companies that are using traditional methods because of needed extra efforts and training. Hence, it is important to provide strong incentives for enterprises for adoption of new technologies.

3.7 Conclusions

The proposed framework provides an implementation approach for protecting data stored on RFID tags, which are based on RBAC and multilevel encryption. The approach covered the conceptual system design and elaborated the interaction between the system components. In addition, a more accurate definition of simple/complex XACP is given, together with an algorithm for determining the type of a specific XACP. Moreover, we presented the first workable cryptographic-based method that can handle complex XACP cases without helps from an online trusted server. It is done through transforming the original document and converting the complex XACP problem to a simple XACP problem.

To clarify the introduced concepts, a conceptual data structure with a sample XML file were used as an example. The proposed conceptual data structure provided a structured approach for managing the memory of RFID tags. To better understand the framework, all the processes involved in encryption and decryption are explained explicitly with an example. To reduce the file size after XML encryption, two new encryption modes were

discussed and difference in size were shown in experimental results.

Furthermore, the potential challenges to the scope and implementation of the framework were identified and discussed briefly.

Chapter 4

Case Study

4.1 Introduction

In this chapter, we present the case study implemented at Concordia University facilities to test our proposed framework with ongoing RFID based FM and emergency management research projects. The FM project facilitates the process of progress monitoring and status tracking of facilities components during their lifecycle. RFID tags are used to store maintenance, inspection and environment information of a component and its surrounding area where it is deployed. The information stored on RFID tags serves various purposes for different operations. In this case study the main focus is on safety equipments that play a crucial part in case of emergencies like fire, but we also take into consideration the information about various conditions at different parts of building, such as hazard materials being used in certain labs, people using lab and current temperature. Though the current projects increase the efficiency of the system but they suffer greatly from security issues, especially, access control.

Different parts of data stored on RFID tags must be accessible to authorized users as per their permissions. For example, an inspector has permission to update some parts of data and information such as hazard-material should be available to fire fighters and safety

personals only. However, current approaches do not cope with this limitation and once a user login to the system, he can access all the information stored, weather he has permission or not.

To remove security problems, we introduce our security framework with RBAC. The required software module are designed and implemented.

4.2 Case Study: Facilities Management & Emergency Response

4.2.1 Background of the Case Study

Concordia Environmental Health and Safety Office (EH&S) is in charge of operation and maintenance of a big and highly dispersed environment. It covers an operational area of 75 buildings with more than 3500 fire extinguishers and 220 fire valves which need to be inspected on a regular basis. The huge amount of effort and investment need to be considered in order to operate and maintain such an environment. The conventional process of inspection, test and maintenance of the safety equipments is operational but it is not efficient, does not cover security aspects and can be improved using process improvement techniques and also by introducing new emerging technologies [53].

For the purpose of fire safety equipments management at Concordia University, Concordia EH&S office provides set of requirements that are: (1) Easy identification of equipments, (2) Structured documentation, (3) Decentralized data storage, (4) Fraud prevention, (5) Paperless information management, (6) Reduce human resource and increase activities efficiency, (7) Standard compliance, and (8) Costs Reduction. Based on the assessment of the requirements and considering available resources and technologies, we have identified that introduction of our RFID security framework in the existing system can fulfill

most of the above-mentioned requirements and provides us with a real-world case study to implement our proposed framework.

In this case study, three roles are used: inspector, fire fighter and general user. All the three roles have different permissions and work at different security levels. Although our main objective is to provide access control in the proposed RFID based approach, we also investigate the scope of adding additional information like floor plan on RFID tags. RFID tags are used for storing information about fire safety equipments and building environment and various components within that environment. Amongst these equipments, fire extinguishers are chosen because of their importance and the higher frequency of their maintenance activities, but this can be easily expanded to other types of equipments as per there importance and usage for different applications. In this case study, all the required software modules have been developed and the applicability of the RFID based system has been tested by several field experiments.

In case of emergencies, to find fire extinguishers, the fire fighters take the arrival instruction for that alerting line and immediately start for the building in which the alarm has been activated. The arrival instruction contains only little information about the building. Usually only the building number, the number of the fire detection line, and the way to the fire brigade terminal are drawn on a small map [66]. Once the respective building is reached a central fire alarm station displays the fire detection line and exact fire detector that has caused the alarm. Building maps at these terminals are available in paper format.

Insufficient information about orientation and reconstruction work within a building often make it very difficult to find the exact fire extinguisher. Huge amount of effort is required to keep these building maps up-to-date and distribute them to different central fire stations. The orientation within a building, information about the direct way to the fire extinguisher, the exact position of the fire detector, the areas where employees and students are located are very crucial to save human lives [66].

Fire extinguishers should be regularly inspected, maintained, recharged and tested based on National Fire Protection Association (NFPA) [56] regulations and guidelines. Strict safety regulations of sensitive systems in buildings, such as fire related subsystems, force the owners to spend huge amount of money to perform inspection on a regular basis. Also, only approved fire inspector should do inspection and the information must be kept confidential from unauthorized personals.

The results of inspection and maintenance should be stored securely and made available to owners and fire fighters only. All the records regarding confidential building information like use of hazard material in certain labs are kept in a sealed envelope and in case of fire emergencies, this information is given to fire fighters at the spot, this further adds extra time to already cumbersome process.

Case Study Target Components

We considered components which could prove helpful in case of fire. Portable fire extinguishers are intended as a first line of defense to cope with fires of limited size. The extinguishers are available in all Concordia buildings and their inspection and maintenance are of high importance.

RFID tags could also be deployed in laboratories which contain hazard material. These tags contain information about the material used and the people working in the specific facility. Currently the information about hazard-materials is available in paper format floor plan, available only at time of emergency, causing delay in rescue process. Also, this floor plan does not provide any information about number of people works in the given vicinity.

4.2.2 Existing Procedures

The service technicians receive their daily work assignment on paper. The work assignments consist of a list of fire extinguishers to be inspected and the respective maintenance

orders. First, they have to find and access each extinguisher, which might be difficult to get at. In order to facilitate locating the fire equipments, inspectors are provided with paper-based floor plans with fire equipment signs. Figure 4.1 show floor plan with fire extinguisher signs. Similarly, fire fighters are given paper based floor plan with extra information, because of safety regulations we cannot show that floor plan. Finding equipments are time consuming process since the paper maps are not easy to read and not regularly updated. A floor plan with extra information contains confidential information and must be kept secret from unauthorized users.

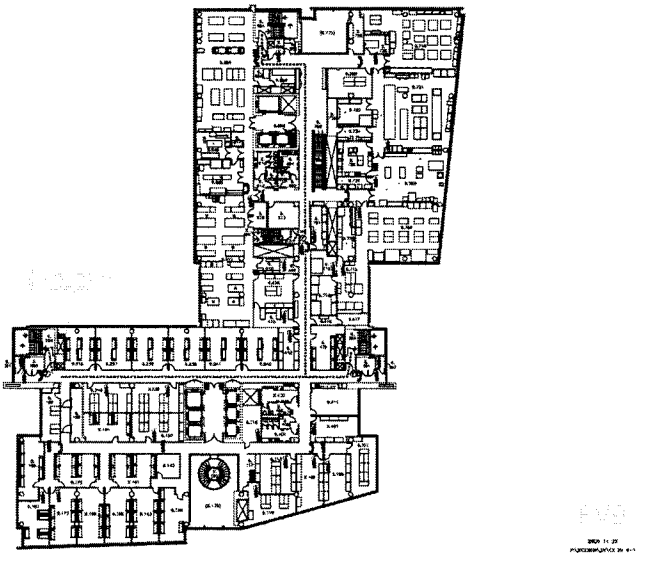


Figure 4.1: Floor Plan with Fire Equipments Signs

Before using tablet PC for manual inspection data entry, the technicians handwrote lengthy maintenance reports after inspection as proof that they had actually done the inspection. Later, they handed in completed maintenance reports to the office clerks, who in turn entered the report into the back-end system and forwarded it for archiving. In the case of defects, an overhaul had to be planned. This paper-based process proved to be error prone, and manual data entry is time-consuming. With the paper-based archive, obtaining

detailed information on actual maintenance history was a tedious process.

Because maintenance reports were often poorly structured or information was missing, many additional inquiries arose. This further adds to the cost of whole process. Concordia hires contractors to perform most inspections and pays them on the basis of the number of extinguishers they inspect. In the past, Concordia EH&S office had to control the work that has been done by the contractor to verify the quality of the service. A newer system has been adopted by the external contractor. In the new system the handheld devices are used to scan bar-codes that are attached to both safety equipments and also the location of the items (fire extinguishers cage or fixtures) [53].

4.2.3 Proposed System

In our prototype system, RFID tags are attached to various building components. Information required for various tasks is stored on the tags. By storing information on the tags itself, used in various processes such as inspection and maintenance would integrate several operations. This helps in achieving decentralized data storage and eliminating the role of central database. RFID tags acts as mini databases and become a potential target of attack. An extra layer of protection is required on the information stored in tags.

Information stored in RFID tags has different security levels, some sections of information do not require any protection, i.e., should be available to everyone; but some parts are very sensitive and must be revealed to specific roles. For example, inspector and maintenance/ repair personnel have access to the information about the history and the condition of the extinguishers, and information about location and floor plans is assessable to all users, without having access to any central database. Thus, it provides data redundancy and eliminates the rework due to not-up-to-date data.

Encryption is used to protect data from the adversaries and unauthorized users. One key is provided to each role to decrypt part of the document, corresponding to its permissions.

Key is transmitted through secure channel. The process of encryption was explained with technical details in Section 3.3. In this case study, we conducted tests with three different roles; inspector, fire fighter and general user. The whole document is encrypted and each role has a separate key to access its share of information, this provides fraud prevention and increases the reliability.

Easy identification of components is also achieved by developed Graphical User Interface (GUI) that shows scanned information from RFID tags attached to various components on the floor plan. The information about part of floor plan is also stored on RFID tags, this helps in generating the floor plan dynamically. This dynamic floor plan generation eliminates many limitations of paper based floor plan. Our prototype takes into consideration most of the requirements specified by Concordia EH&S office. One of the requirements of cost reduction is arguable but with RFID technology getting cheaper every year, this will be achieved in coming years as well. The primary tests were performed on active tags with 32 KB of memory. Moreover, the selected tags are designed to work well near liquids and metals.

Conceptual System Design

Facility Management & Emergency Response System is initially composed of following components:

1. *RFID Tags*: Active tags attached to fire extinguishers that contains information about the extinguisher.

2. *RFID Reader*: PCMCIA portable reader module compatible with proprietary active tags that is being held by Concordia EH&S personnel.

3. *Handheld Device*: PDAs equipped with PCMCIA slot that host the RFID reader and the software.

4. *Software*: the developed software for communicating with RFID tags and readers.

The conceptual diagram in Figure 4.2 illustrates the design of the RFID based system. It shows the interaction between the hardware and software components as well as the interaction of the end-user with the system by the means of the GUI. Furthermore, the diagram lists the functionalities available to the end-user which include: reading from the tag, performing the inspection, updating the tag, writing to the tag, etc. The software controls the reader to start scanning and display the received tags. To perform an inspection, the user has to select the inspector role and provide his role key. After role and key verification, user can perform inspection and the new information will be written on the tag.

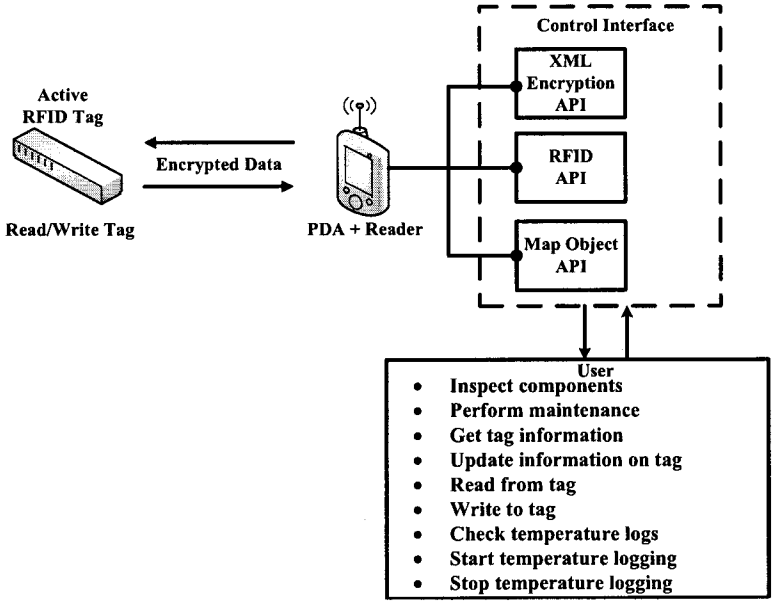


Figure 4.2: Design of RFID Based Facilities Management and Emergency Response System

RFID Tag Data Structure

The designed system is intended to work for different types of components and different tag types. In other words the design aims to be expandable and support multi-tag environment.

In the case study different tags with various memory sizes were used. We conducted experiments with active tags with 8 KB and 32 KB memory. Active tags with 32 KB memory also has embedded sensors, that is used to measure the temperature. Different amount of data is being stored in RFID tags, depending on their memory and location of deployment.

Based on our design, extra information about the component and its surrounding other than its unique ID is needed to be written on the tag. The required data fields should be carefully selected based on the requirements, in case of low memory active tags the data should be "abbreviated" to smaller set and stored on the tags. Active tag with large memory contains location/floor plan data and information needed for inspection and emergency scenarios.

Selection of crucial data to be stored on the extinguisher tags are based on the NFPA guidelines and result of the meetings with field inspectors and EH&S staff [53]. The most important criteria for selecting the data field are the applicability of data. The main purpose of adding data as well as the ID on the tags is to provide the capability to perform information gathering/maintenance/repair activities without having access to data in database.

The memory of the tags has been segmented and contains to the following information [53]: (1) ID, (2) Specification (e.g., manufacturing date), (3) Status, (4) Maintenance data (e.g., condition and defective part), (5) History (e.g., last inspection date) and, (6) Environment data (e.g., location).

For the use in the BIM-based indoor-emergency-navigation-system, a data format is needed which contains the information in a unique structure to use only one tool for displaying the plans on mobile devices or generating routing maps. Defined by Industry Foundation Classes (IFC), the IFC-XML is an exchange format for building models. It has been specified by the IAI (International Alliance for Interoperability). It became an international standard and defines an exchange format and contains object classes for storey, roofs, walls, stairs, etc. [66].

In earlier experiments, we stored parts of the floor plan in image format such as JPG, on different RFID tags. This is very crude and static way of representation; neither has it helped in locating components nor in navigation. To enhance the system functionality, image format is replaced with CAD file format. Though CAD file represent floor plan in detail but the size of CAD files is very large, even the part of full floor plan is too big to fit. In current experiments we stored the CAD file on the handheld device, but we are already working to dynamically draw the floor plan from the information stored on RFID tag using IFC-XML, instead of image or CAD files.

In active tags all the information is stored in XML format. A new XML file is created based on the NFPA guidelines, part of this file is described in Figure 4.3.

Prototype Software Flowchart

The proposed flowchart extends [53] work in facilities management project, by adding different roles and security levels. The user logs in to the system in order to proceed with the software. The login procedure involves entering the username (code) and the password. The EH&S users would be assigned an RFID badge that has their user code. The inspector or fire fighter can login to system by scanning his/her RFID badge and entering the password. The system can support fingerprint authentication in conjunction with ID scan in future improvement. The username and password are different from role and its corresponding key. This extra layer of security is added to differentiate between different users within a same role. This would guaranty that only authorized user is logging to the system and accessing only permitted portion of the document. This could be used as a fraud prevention feature.

After authentication, the user would have access to certain operations which are common for all roles. These operations include; view tag information and check current temperature of the surrounding environment of scanned tag. Figure 4.5 shows a snapshot of

graph showing temperature retrieved from the RFID tag. This temperature information is stored in a log file on the tag itself, and is not a part of the XML file. The data structure of temperature log file is vendor dependent. After selecting a valid tag index, the software loads general operations and any user can retrieve very limited information about specific tag. To access more information user has to select its role and provide a valid key. Once key is validated, prototype loads role specific operations. Figure 4.4 shows software flowchart.

The major operations developed for prototype system are: (1) new maintenance task and (2) view data. New maintenance task is designed for FM inspector and view data for fire fighter. After selecting a role and providing valid key, EH&S worker can load the assigned *job*. *Job* is the list of inspection/maintenance activities that is planned for the inspector by supervisor. The inspector can manually add extinguisher IDs that are not in the list but are needed to be added to the task list. Software would provide visualization of the extinguishers in the job list on the floor plans to aid the inspector locate the extinguishers on the floor.

The software also provides navigation aid for the inspector to locate the extinguishers in the building using active tags. The software has pre-loaded floor plans as a visualization layer. In these experiments we used CAD file to display floor plan pre-loaded in the hand-held devices. The floor plan created from the information based on IFC standard, stored in RFID tag is not very clear and does not contain all the details yet. To create a full floor plan with all the details using IFC standard is a part of future work. By surveying the area to detect the tags, the sensed tags are shown on the floor plan based on their location information. Using the long range tags allows inspectors to visually find the surrounding components on the floor plan and decreases the time needed to locate them. Figure 4.6 shows a sample snapshot of the screen, where the locations of sensed components are shown with green stars and the selected RFID tag is shown red square shape on the floor plan and it helps the user to visually identify the components on the plan. Figure 4.6 also shows scanned

tags and two types of information. The first type is general information, it is not stored in XML file and can be accessed by any role (shown in green rectangle), the second type is confidential information (shown in red rectangle), stored in XML file and can only be seen after decrypting with either inspector or fire fighter role key.

After finding the extinguisher, the inspector starts the task by scanning the extinguisher tag. It might be possible that more than one tag is being read while scanning, in this case, the system shows the information of the sensed tag and the user is required to choose the appropriate tag to proceed using an indexing system. Once the tag is selected, software shows its location on the map by collection information from the tag. Before the other information related to inspection is shown, user has to select role and provide valid key. The data that is recorded on the tag is shown and it helps the inspector to quickly review the history of inspection and also the previous maintenance result and information about the type and possible defects of the extinguisher.

Before performing the actual inspection, the software automatically generates alerts for the inspector based on the data that is available on the extinguisher tag. The alerts are designed to warn the inspector about the required maintenance/repair/replacement procedure based on the regulation.

The software contains the checklist of inspection and maintenance activities based on the type of extinguishers. It can include the standard procedures as a reference for less experienced EH&S workers. The inspector performs the tasks and complete easy-to-fill forms (in the format of checkboxes or drop down menus). This data entry step is comparatively faster and more accurate than paper based system. Furthermore, the data will be saved in a structured fashion and and well protected form the adversaries.

The inspector views the result of inspection/maintenance activity and confirms. Before the software updates the data on the tag, it encrypts the modified data with inspector role key. The task is considered to be completed only after the successful data update of RFID

tag. Flowcharts of view data and new management task are shown in Appendix A.

Hardware Test

Several hardware-related tests have been conducted in order to identify appropriate RFID tags and readers for the case studies. The test has been designed to identify suitable set of tags and readers and antennas with required readability range, ruggedness, and noise protection for each scenario and component.

4.3 Summary

This case study has been done in the EV building of Concordia University where active tags were attached to 9th floor fire extinguishers. The technological feasibility of the system has been tested in a real working environment.

The case study showed the applicability of the proposed approach. This covered different types of components and facilitated different processes. As efforts to develop full secure RFID communication model for facilities management and emergency response is still evolving, the full implementation of the case study should be considered as the first step. It is concluded that further study of XML data format for representing safety equipments and floor plan using IFC-XML format are necessary for future improvements.

```
<Facility-Management-Emergency-Response-System>
    <Environment>
        <Location>
            <Building></Building>
            <Floor></Floor>
            <Room></Room>
        </Location>
        <Map>
        <Nodes>
            <Node Number= "">
                <x></x>
                <y></y>
            </Node>
        </Nodes>
        <Links>
            <Link Number="">
                <Start-Node></Start-Node>
                <End-Node></End-Node>
            </Link>
        </Links>
        </Map>
        <Hazard-Material>
            <Type></Type>
            <Room-Incharge></Room-Incharge>
        </Hazard-Material>
    </Environment>
    <ID>
        <Type></Type>
        <Model></Model>
        <Serial></Serial>
    </ID>
    <Specification>
        <Manufacturing-Date></Manufacturing-Date>
    </Specification>
    <Status>
    </Status>
    <Maintenance-Data>
        <Condition>
            <Obstructed></Obstructed>
            <Pressure></Pressure>
            <Loose></Loose>
            <Dusted></Dusted>
            <Rusted></Rusted>
            <Damaged></Damaged>
        </Condition>
        <Defective-Part>
            <Missing-Pin></Missing-Pin>
            <Missing-Rivet></Missing- Rivet>
            <Missing-Label></Missing- Label>
            <Missing-Sign></Missing- Sign>
            <Neck-Bended></Neck-Bended>
            <Plugged-Hose></Plugged-Hose>
            <Seal-Broken></Seal-Broken>
        </Defective-Part>
    </Maintenance-Data>
    <History>
        <Last-Inspection-Data></Last-Inspection-Data>
        <Comments></Comments>
        <Inspector-Name></Inspector-Name>
    </History>
</ Facility-Management-Emergency-Response-System >
```

Figure 4.3: Revised XML File for Active Tag Data Structure

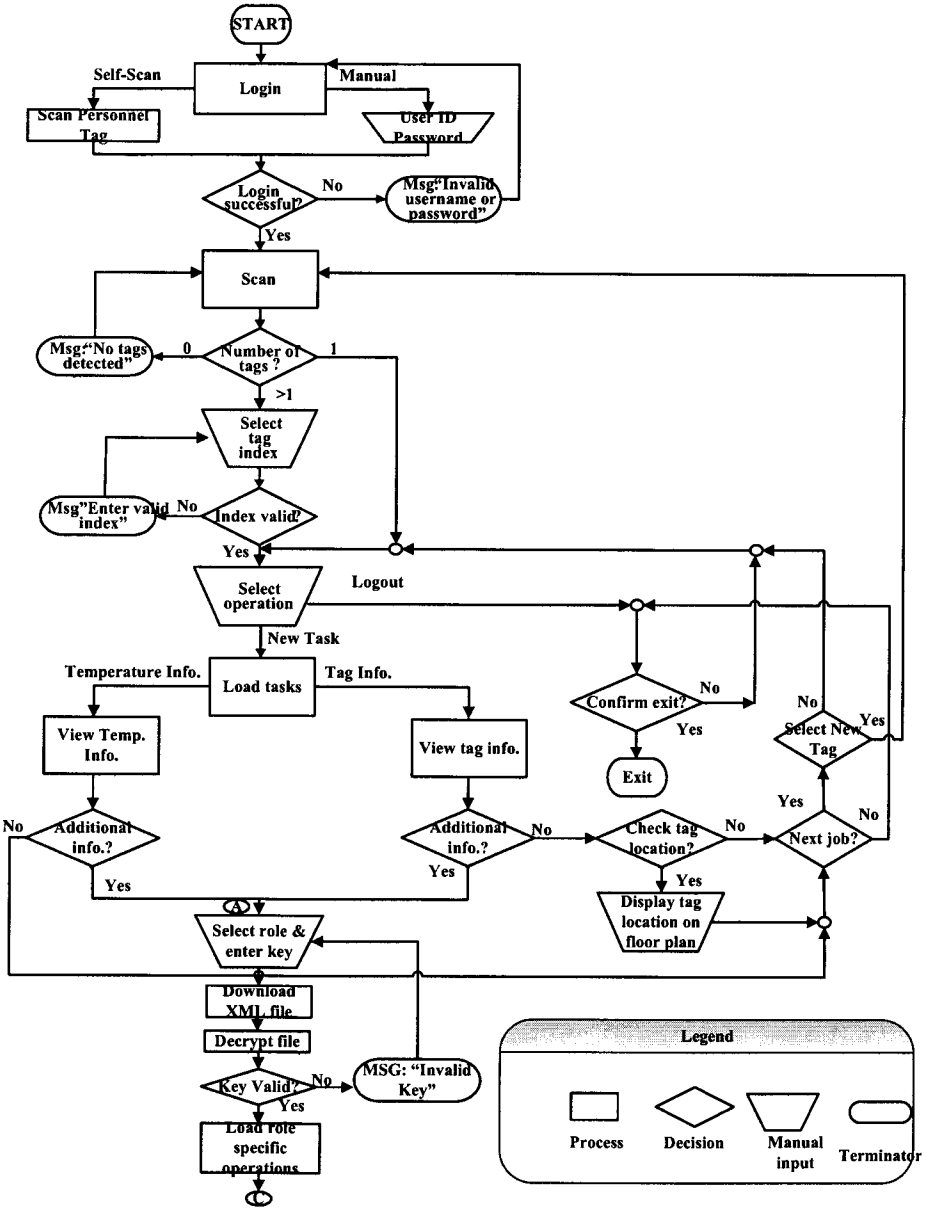


Figure 4.4: Software Flowchart

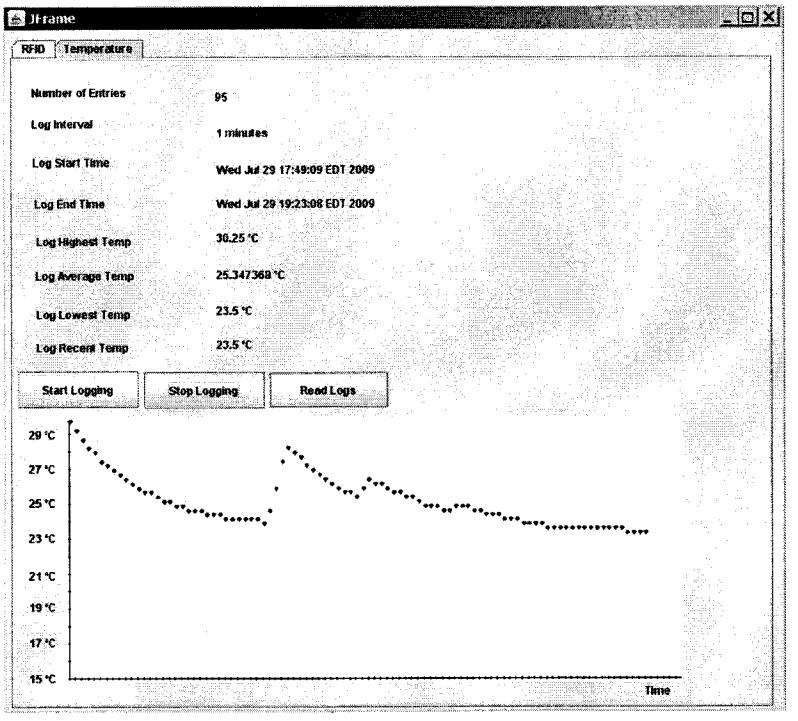


Figure 4.5: Snapshot of Temperature Aware Software

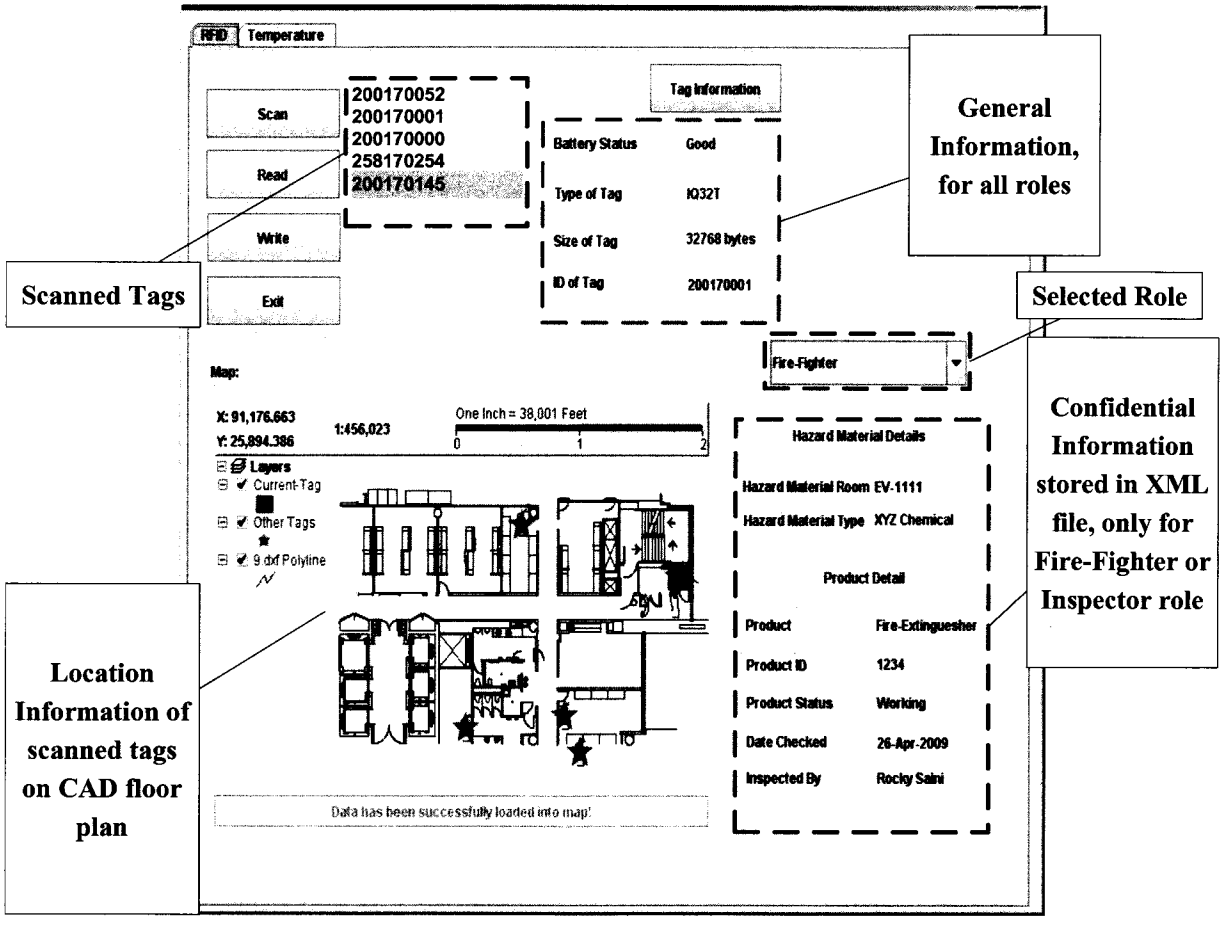


Figure 4.6: Snapshot of Location Management Software with Confidential and General Information

Chapter 5

Conclusions and Future Work

5.1 Research Summary

In this research, we highlighted that there is an increasing interest in using RFID tags as distributed storage and proposed the first work that investigates security issues, in particular access control, in this new type of applications. A framework to protect data on the RFID tags through RBAC is proposed. Users are assigned different roles as per there permissions; this also helps in managing large number of users. Multilevel encryption secures different parts of same document from adversaries, and users of the system having different permissions. In this research, simple and complex XACPs are discussed thoroughly and more profound and accurate definition is given along with an algorithm to draw a differentiating line between them. The proposed access control model is very flexible and can accommodate both simple and complex ACPs. We introduced the first cryptographic solution to protect the data stored on the RFID tags without doing any cryptographic computations on the tag itself.

The choice of XML format was made because of its acceptance as a standard to store and exchange information in vast number of applications. Moreover, XML encryption specifications are well defined and standardized. Although the proposed framework and

algorithms are defined using XML format, it can be readily extended to support other types of data representations with hierarchical structure.

The given framework enhances the scope of existing RFID base FMS by providing security with RBAC and multi-level encryption. Our research provides a secure communication framework to support a futuristic vision of RFID based distributed storage environment. Although the case study showed the technical feasibility of our proposed framework using available hardware, several challenges identified in this research should be addressed to make the vision feasible.

5.2 Research Contributions and Conclusions

The introduced security framework ensures confidentiality and integrity of the data stored on the RFID tags in distributed storage applications. This would address the need of security by utilizing multi-level encryption together with RBAC, which has been identified as a major challenge in previous related research. The proposed algorithms cover both simple and complex XACPs and successfully transform a complex policy into a simple one to accommodate broad range of ACPs, which are very common in real-world applications and have never been addressed thoroughly before.

Our research is similar to secure data publishing of XML document and does not depend on the computational capabilities of RFID tags. The proposed framework can work efficiently with the limited computational capabilities of RFID tags to perform read and write operations. The case study showed the applicability of our proposed approach and validated the usability of the various methods identified in this research.

The research contributions can be summarized as follows: (1) Provide a secure framework that ensures data confidentiality and integrity, with multi-level encryption and RBAC of the data stored on an RFID tag; (2) A more systematic definition of simple and complex XACPs is given, that it helps in selecting storage-efficient encryption method; (3) Explore

various modes to encrypt XML documents, apart from standard specification, in order to save storage space; (4) Introduce a workable cryptographic solution to handle complex XACPs; and (5) Demonstrate a case study to verify the applicability of the proposed approach.

5.3 Future Work

The following steps are necessary for fully realizing the proposed approach: (1) To develop a protocol that manages the updates in XACPs; (2) To design an algorithm that generates role keys in accordance with the changes in XACPs and XML document structure; (3) To introduce techniques that can be used to implement specific type of access, such as read/write access; and (4) To investigate various real world applications to test our proposed approach.

References

[1] Selim G. Akl and Peter D. Taylor. Cryptographic solution to a problem of access control in a hierarchy. *ACM Transactions on Computer Systems (TOCS)*, 1(3):239–248, 1983.

[2] Ross J. Anderson and Markus G. Kuhn. Low cost attacks on tamper resistant devices. In *Proceedings of the 5th International Workshop on Security Protocolsm, LNCS 1361*, pages 125–136, 1997.

[3] Giuseppe Ateniese, Jan Camenisch, and Breno de Medeiros. Untraceable RFID tags via insubvertible encryption. In *Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS '05)*, pages 92–101, 2005.

[4] P. Bahl and V. N. Padmanabhan. Radar: an in-building rf-based user location and tracking system. volume 2, pages 775–784, 2000.

[5] Elisa Bertino, Barbara Carminati, and Elena Ferrari. Access control for XML documents and data. *Information Security Technical Report*, 9(3):19–34, 2004.

[6] Manish Bhuptani and Shahram Moradpour. *RFID Field Guide: Deploying Radio Frequency Identification Systems*. Sun Microsystems Press, New Jersey, 2005.

[7] Chia-Chen Chao, Jiann-Min Yang, and Wen-Yuan Jen. Determining technology trends and forecasts of RFID by a historical review and bibliometric analysis from 1991 to 2005. *Technovation*, 27(5):268–279, 2007.

[8] Vipul Chawla and Dong Sam Ha. An overview of passive RFID. *IEEE Communications Magazine*, 45(9):11–17, 2007.

[9] Eun Young Choi, Su Mi Lee, and Dong Hoon Lee. Efficient RFID authentication protocol for ubiquitous computing environment. In *Proceedings of the First International Workshop on Security in Ubiquitous Computing Systems (EUC), LNCS 3823*, pages 945–954, 2005.

[10] Jason Crampton. Applying hierarchical and role-based access control to XML documents. In *ACM Workshop on Secure Web Services*, pages 37 – 46, 2004.

[11] Jason Crampton, Keith Martin, and Peter Wild. On key assignment for hierarchical access control. In *CSFW '06: Proceedings of the 19th IEEE workshop on Computer Security Foundations*, pages 98–111, Washington, DC, USA, 2006. IEEE Computer Society.

[12] Ernesto Damiani, Sabrina De Capitani Di Vimercati, Stefano Paraboschi, and Pierangela Samarati. A fine-grained access control system for XML documents. *ACM Transactions on Information and System Security (TISSEC)*, 5(2):169–202, 2002.

[13] Kurt Dittmer. Blue force tracking a subset of combat identification, Military Review, 2004.

[14] Konstantinos Domdouzis, Bimal Kumar, and Chimay Anumba. Radio-frequency identification RFID applications: A brief introduction. *Adv. Eng. Inform.*, 21(4):350–355, 2007.

[15] EPCglobal. http://www.epcglobalinc.org/.

[16] EPCglobal. Specifications of rfid air interface. Retrieved from http://www.epcglobalinc.org/standards/uhfc1g2/uhfc1g2_1_1_0-standard-20071017.pdf on December 5, 2008.

[17] Esin Ergen, Burcu Akinci, Bill East, and Jeff Kirby. Tracking components and maintenance history within a facility utilizing radio frequency identification technology. *Journal of Computing in Civil Engineering*, 21(1):11–20, 2007.

[18] Esin Ergena, Burcu Akinci, and Rafael Sacks. Life-cycle data management of engineered-to-order components using radio frequency identification. *Advanced Engineering Informatics*, 21(4):356–366, 2007.

[19] Martin Feldhofer. An authentication protocol in a security layer for RFID smart tags. In *Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference (MELECON)*, pages 759–762, 2004.

[20] David F. Ferraiolo, D. Richard Kuhn, and Ramaswamy Chandramouli. *Role-Based Access Control*. Artech House Publishers, 2003.

[21] T.A.S. Foundation. Apache xml security java. http://projects.apache.org/projects/xml_security_java.html.

[22] Bernard Ghanem. RFID project technical report. http://vision.ai.uiuc.edu/bghanem2/Files/RFID%20Tech%20Report.pdf.

[23] Aim Global. What is RFID. http://www.aimglobal.org/technologies/rfid/what_is_rfid.asp.

[24] Howard Gobioff, Sean Smith, J.D. Tygar, and Bennet Yee. Smart cards in hostile environments. In *Proceedings of the Second USENIX Workshop on Electronic Commerce*, 1996.

[25] Nathan Good, John Han, Elizabeth Miles, David Molnar, Deirdre Mulligan, Laura Quilter, Jennifer M. Urban, and David Wagner. Radio frequency id and privacy with information goods. In *Proceedings of ACM Workshop on Privacy in the Electronic Society (WPES)*, pages 41–42, 2004.

[26] Associated General Contractors Guide. The contractor's guide to bim. http://www.agcnebuilders.com/documents/BIMGuide.pdf.

[27] Mark Harrison. Guidelines for lifecycle ID & data management. Technical Report AEROID-CAM-014, Auto-ID Lab, University of Cambridge, August 2007.

[28] Mark Harrison and Ajith Kumar Parlikad. Lifecycle ID and lifecycle data management. Technical Report WP-BIZAPP-032, Auto-ID Lab, University of Cambridge, June 2006.

[29] Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster. The anatomy of a context-aware application. In *Mobile Computing and Networking*, pages 59–68. ACM Press, 1999.

[30] Mike Hazas, James Scott, and John Krumm. Location-aware computing comes of age. *Computer*, 37(2):95–97, 2004.

[31] Claus Heinrich. *Global Positioning System: Theory and Practice, Fourth Edition.* Springer-Verlag, 1997.

[32] Jeffrey Hightower and Gaetano Borriello. A survey and taxonomy of location systems for ubiquitous computing. Technical report, IEEE Computer, 2001.

[33] B. Hofmann-Wellenhof, Herbert Lichtenegger, and James Collins. *RFID and Beyond: Growing Your Business Through Real World Awareness.* Wiley Publishing, Indianapolis, IN, 2005.

[34] Sozo Inoue and Hiroto Yasuura. RFID privacy using user-controllable uniqueness. In *RFID Privacy Workshop*, 2003.

[35] Umit Isikdag, Jason Underwood, and Ghassan Aouad. An investigation into the applicability of building information models in geospatial environment in support of site

selection and fire response management processes. *Adv. Eng. Inform.*, 22(4):504–519, 2008.

[36] RFID Journal. The history of RFID technology. http://www.rfidjournal.com/article/view/1338/1.

[37] Ari Juels. Minimalist cryptography for low-cost RFID tags. In *Proceedings of the 4th International Conference on Security Communication Networks, Springer, LNCS 3352*, pages 149–164, 2005.

[38] Ari Juels. Strengthening EPC tags against cloning. In *Proceedings of the 4th ACM Workshop on Wireless Security (WiSE'05)*, pages 67 – 76, 2005.

[39] Ari Juels. RFID security and privacy: A research survey. *IEEE Journal on Selected Areas in Communications*, 24(2):381–394, February 2006.

[40] Ari Juels, Ronald L. Rivest, and Michael Szydlo. The blocker tag: Selective blocking of RFID tags for consumer privacy. In *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03)*, pages 103–111, 2003.

[41] Ari Juels, Paul Syverson, and Dan Bailey. High-power proxies for enhancing RFID privacy and utility. In *Proceedings of the 5th Workshop on Privacy Enhancing Technologies (PET)*, pages 210–226, 2005.

[42] Günter Karjoth and Paul A. Moskowitz. Disabling RFID tags with visible confirmation: Clipped tags are silenced. In *Proceedings of the 2005 ACM Workshop on Privacy In The Electronic Society (WPES'05)*, pages 27 – 30, 2005.

[43] Esin; Tang Pingbo Kiziltas, Semiha; Akinci Burcu; Ergen. Technological assessment and process implications of field data capture technologies for construction and facility/infrastructure management. *ITcon*, 13:134–154, 2008.

[44] Carl Lake and Edward J. Jaselskis. RFID applications for owners and contractors. In *Proc. Construction Industry Institute Annual Conference*, pages 7–24, 2000.

[45] J. Landt. The history of RFID. *Potentials, IEEE*, 24(4):8–11, 2005.

[46] Marc Langheinrich. FragDB – secure localized storage based on super-distributed RFID-tag infrastructures. In *Proceedings of International Conference on Mobile Data Management*, pages 233–237, 2007.

[47] Berin Lautenbach. Introduction to xml encryption and xml signature. *Information Security Technical Report*, 9(3):6–18, 2004.

[48] Shih-Wei Lee, Shao-You Cheng, Jane Yung-Jen Hsu, Polly Huang, and Chuang-Wen You. Emergency care management with location-aware services. In *Proceedings of Pervasive Health Conference and Workshops*, pages 1–6, 2006.

[49] Mats G. Lindquist. RFID in libraries - introduction to the issues. In *Proceedings of the 69th IFLA General Conference and Council*, 2003.

[50] T. McCoy, R.J. Bullock, and P.V. Brennan. RFID for airport security and efficiency. Oct. 2005.

[51] Gerome Miklau and Dan Suciu. Controlling access to published data using cryptography. In *Proceedings of the 29th International Conference on Very Large Data Bases (VLDB)*, pages 898–909, 2003.

[52] Jeffrey C. Mogul and Martin Arlitt. SC2D: An alternative to trace anonymization. In *Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data (MineNet '06)*, pages 323 – 328, 2006.

[53] Ali Motamedi. Framework for lifecycle management of facilities components using rfid technology. Master's thesis, Concordia University, 2009.

[54] Ali Motamedi and Amin Hammad. Lifecycle management of facilities components using radio frequency identification and building information model. *ITcon*, 14:238–262, 2009.

[55] Tomasz Müldner, Gregory Leighton, and Jan Krzysztof Miziołek. Using multi-encryption to provide secure and controlled access to XML documents. In *Proceedings of Extreme Markup Languages 2006*, 2006.

[56] NFPA. National fire protection association, 2008. http://www.nfpa.org.

[57] L. M. Ni, Yunhao Liu, Yiu C. Lau, and A. P. Patil. Landmarc: indoor location sensing using active RFID. pages 407–415, 2003.

[58] US National Institute of Building Sciences. National building information modelling standard, part-1: Overview, principles and methodologies, December 2007. http://www.wbdg.org/pdfs/NBIMSv1_p1.pdf.

[59] Radu-Ioan Paise and Serge Vaudenay. Mutual authentication in RFID: Security and privacy. In *Proceedings of ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, pages 292–299, 2008.

[60] VDC research. RFID: Acceleration from 2008 will meet recession in 2009, 2009. http://www.vdcresearch.com/Default_temp.asp.

[61] Keunwoo Rhee, Jin Kwak, Seungjoo Kim, and Dongho Won. Challenge-response based RFID authentication protocol for distributed database environment. In *Proceedings of International Conference on Security in Pervasive Computing (SPC), LNCS 3450*, pages 70–84, 2005.

[62] Melanie R. Rieback, Bruno Crispo, and Andrew S. Tanenbaum. RFID guardian: A battery-powered mobile device for RFID privacy management. In *Proceedings of the*

10th Australasian Conference on Information Security and Privacy (ACISP 2005), 2005.

[63] Melanie R. Rieback, Bruno Crispo, and Andrew S. Tanenbaum. Is your cat infected with a computer virus? In *Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM 06)*, 2006.

[64] Melanie R. Rieback, Bruno Crispo, and Andrew S. Tanenbaum. RFID malware: Truth vs. myth. *IEEE Security & Privacy*, 4(4):70 – 72, 2006.

[65] George Roussos. *Networked RFID Systems, Software and Services*. Springer, 2008.

[66] Uwe Rüppel and Kai Stübbe. BIM based indoor-emergency-navigation-system for complex buildings. *Tsinghua Science and Technology Journal*, 13:362–367, October 2008.

[67] Sanjay E. Sarma, Stephen A. Weis, and Daniel W. Engels. RFID systems and security and privacy implications. In *Proceedings of the 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2002)*, pages 1–19, 2002.

[68] Sung-Tsun Shih, Kunta Hsieh, and Pei-Yuan Chen. An improvement approach of indoor location sensing using active rfid. In *ICICIC '06: Proceedings of the First International Conference on Innovative Computing, Information and Control*, pages 453–456, Washington, DC, USA, 2006. IEEE Computer Society.

[69] Derek Smyth. XML encryption, December 2006. http://dotnetslackers.com/articles/xml/XMLEncryption.aspx.

[70] IDENTEC SOLUTIONS. Active UHF tag i-Q32T. http://www.identecsolutions.com/ilrproductsheets.html.

[71] Jongchul Song, Carl T. Haas, and Carlos H. Caldas. A proximity-based method for locating RFID tagged objects. *Adv. Eng. Inform.*, 21(4):367–376, 2007.

[72] M. J. Spearpoint. Integrating the IFC building product model with zone fire simulation software. In *Proceedings of International Conference on Building Fire Safety*, pages 56–66, 2003.

[73] Harry Stockman. Communication by means of reflected power. *Proceedings of the IRE*, pages 1196–1204, October 1948.

[74] Frank Thornton and Chris Lanthem. *RFID Security: Protect the Supply Chain.* Syngress, 2005.

[75] Gene Tsudik. A family of dunces: Trivial RFID identification and authentication protocols. In *Proceedings of the 7th Privacy-Enhancing Technologies Symposium (PETąŕ07), LNCS 4776*, pages 45–61, 2007.

[76] István Vajda and Levente Buttyán. Lightweight authentication protocols for low-cost RFID tags. In *Proceedings of the Second Workshop on Security in Ubiquitous Computing (Ubicomp)*, 2003.

[77] World Wide Web Consortium (W3C). Extensible markup language (XML). http://www.w3.org/XML/.

[78] World Wide Web Consortium (W3C). XML path language (XPath) version 1.0, November 1999. http://www.w3.org/TR/xpath.

[79] World Wide Web Consortium (W3C). XML encryption syntax and processing, December 2002. http://www.w3.org/TR/xmlenc-core/.

[80] World Wide Web Consortium (W3C). XML-signature XPath filter 2.0, November 2002. http://www.w3.org/TR/xmldsig-filter2/.

[81] World Wide Web Consortium (W3C). XQuery 1.0: An XML query language, January 2007. http://www.w3.org/TR/xquery/.

[82] Roy Want, Andy Hopper, Veronica Falc ao, and Jonathan Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 10(1):91–102, 1992.

[83] Steve H. Weingart. Physical security devices for computer subsystems: A survey of attacks and defenses. In *CHES 2000, LNCS 1965*, pages 302–317, 2000.

[84] Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest, and Daniel W. Engels. Security and privacy aspects of low-cost radio frequency identification systems. pages 201–212. Springer-Verlag, 2003.

[85] Meng Xiao-Feng, Luo Dao-Feng, and Ou Jian-Bo. An extended Role-Based access control method for XML documents. *Wuhan University Journal of Natural Sciences*, 9(5):740–744, 2004.

Appendix A

Appendix - Software Flowcharts

Figure A.1 and A.2 show the flowchart for view data and new maintenance task, respectively. The major processes in software flow (identified by number in flowcharts) are as follow:

1. The user selects a role and enter corresponding key.
2. Software downloads encrypted XML file from the selected tag.
3. Decrypt file using key provided by the user. The partially decrypted portions of XML file are compared with the ACP embedded in the software, if the decrypted content matches with the permitted portions of file, then role specific operations are loaded.
4. After authentication and authorization he selects the type of operation. One of the three major operations could be chosen.
    - *View data*: for retrieval of stored information on the tags such as the history of the objects
    - *New Maintenance*: to start new inspection/maintenance activity
    - *Show Path*: to show a path form one position to another by calculation routing information

5. There are two methods for retrieving information.

    - Scanning tag
    - Entering the ID or the description of the item/location

6. The scanned tag or entered ID, Base on the type of the tag, stored information on the tag is shown.

7. The fire fighter/inspector has the option to view the history of the item/location from the database.

8. For the new maintenance activity there are three options available for the inspector:

9. *Load job*: the task list (the list of fire extinguishers that are planned to be inspected by inspector through operation management system) can be loaded. After choosing this option the list is loaded from the database and the inspector would do the job as planned. After loading the lists the related maps with the target extinguishers are shown in the software.

10. *Scan area*: the inspector have the option to scan the area to detect the location tags. The tags that are sensed by the RFID reader in that area are shown in the related map.

11. *Enter location info*: the inspector chooses this option if he knows which building or floor he wants to inspect. By selecting the desired location from the drop down menus, he selects the location and the related map with the sign of extinguishers is shown in the software.

    After deciding on the inspection target (the extinguisher that is going to be inspected) the inspector would find the item using the map.

12. Inspector starts the inspection/maintenance activity.

13. The location information on the tag is shown. The inspector has the option to edit the data on the location tag, if required.

14. The component specific information is shown.

15. Based on the previous information recorded on the tag, related alerts are shown.

    The sample alerts are: Hydrostatic test needed, Recharging needed, The extinguisher is obsolete and, Extinguisher type is invalid. The alerts are automatically generated by software based on the information on the items and also the time and date on the PC. It would eliminate the chance for human error.

16. A user friendly checklist to log the inspection/maintenance results is shown. The status of the extinguisher and also the name of the defected part can be entered in the system using user friendly interface. The inspector can also get the instruction for inspecting different types of extinguishers if needed.

17. After completing the actual inspection and entering the information about the possible defective part or condition of the extinguisher. The software shows the data to be written in the tag and request for confirmation.

18. If the information is correct the data is encrypted with inspector role key and written on the RFID tag.

19. The inspector can go to the next extinguisher to inspect or terminate the process.

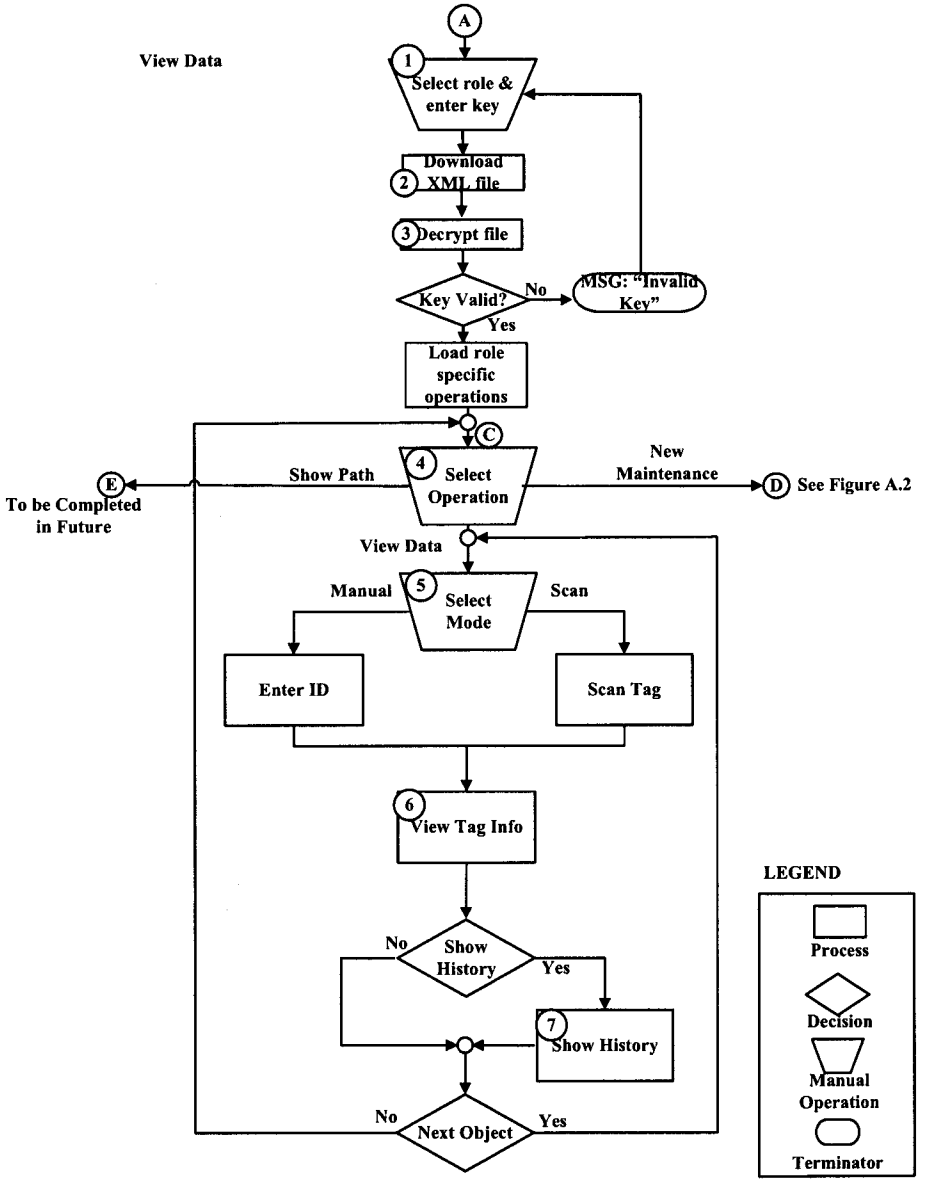


Figure A.1: View Data Flowchart

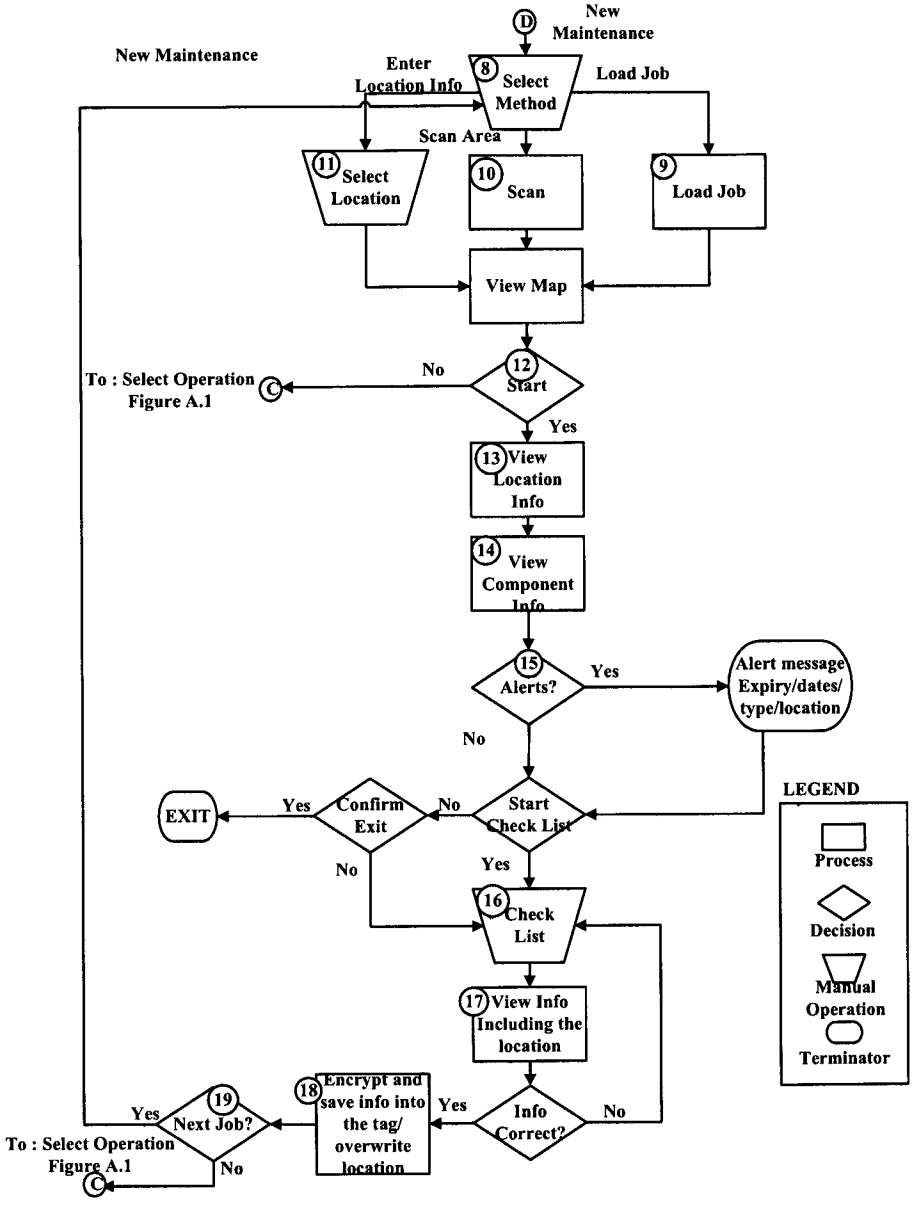


Figure A.2: New Inspection/Maintenance Flowchart