International Journal of Scientific Research in Computer Science, Engineering and Information Technology

© 2017 IJSRCSEIT | Volume 2 | Issue 2 | ISSN : 2456-3307

The Electronic Passport and the Future of Government Issued RFID-Based Identification

V. Suruthi, K. Senthil Kumar

Muthayammal Engineering College, Rasipuram, Rasipuram, Tamil Nadu, India

ABSTRACT

RADIO Frequency Identification (RFID) tags for the function of next-generation electronic product code (EPC) will become one of the most widely used devices in the near future an RFID. Once the tag is found valid, the back-end database will look up its product information for further processing. RFID tags are classified into three types: active, semi passive, and passive. Active tags contain batteries so that they can actively communicate with the reader. Semi passive tags also contain batteries but they wait for the reader's query. As for passive tags, the power comes from the reader. The class of a tag represents the effective reading range. We analyzed the number of rounds required and the period of key update for practical deployment. RFID is an acronym for Radio Frequency Identification. RFID is one member in the family of Automatic Identification and Data Capture (AIDC) technologies and is a fast and reliable means of identifying just about any material object. This project can be used for security purpose where it gives information about the authorized persons and unauthorized persons. This can be applied in real time systems as such in recording the attendance, in the companies, airports for accessing the passports and in industries to know who are authorized. RFID is increasingly used with biometric technologies for security. Primarily, the two main components involved in a Radio Frequency Identification system are the Transponder (tags that are attached to the object) and the Interrogator (RFID reader). Communication between the RFID reader and tags occurs wirelessly and generally doesn't require a line of sight between the devices. RFID tags are categorized as either active or passive. Active RFID tags are powered by an internal battery and are typically read/write, i.e., tag data can be rewritten and/or modified. An active tag's memory size varies according to application requirements; some systems operate with up to 1MB of memory. Passive RFID tags operate without a separate external power source and obtain operating power generated from the reader.

Keywords : RFID, EPC, AIDC, TID, Cyclic Redundancy Check

I. INTRODUCTION

This project uses passive tags. Read-only tags are typically passive and are programmed with a unique set of data (usually 32 to 128 bits) that cannot be modified. The reader has three main functions: energizing, demodulating and decoding. The antenna emits radio signals to activate the tag and to read and write data to it. In this project, the RFID module reader typically contains a module (transmitter and receiver), a control unit and a coupling element (antenna). This module is interfaced with the micro controller and when the card is brought near to the RFID module it reads the data in the card and displays on the LCD. If the data in the card is matched with the data in the program memory then it compares and displays authorized message. If the data is not matched it displays unauthorized. For authorized message, the door gets opened and the person gets in after sometime the door gets closed.

If it is an unauthorized person it alerts the persons through a buzzer. The RFID module indicates a buzzer whenever it reads the data from the RFID card. The significant advantage of all types of RFID systems is the non contact, non-line-of-sight nature of the technology. Tags can be read through a variety of substances such as snow, fog, ice, paint, crusted grime,

CSEIT1722273| Received : 10 April 2017 | Accepted : 20 April 2017 | March-April-2017 [(2)2: 931-935]

and other visually and environmentally challenging conditions, where barcodes or other optically read technologies would be useless.

This project can provide security for the industries, companies, etc. This project uses regulated 5v, 500mA power supply. 7805, a three terminal voltage regulator is used for voltage regulation. Bridge type full wave rectifier is used to rectify the ac output of secondary of 230/12v step down transformer. The RFID module requires a separate +5v power supply.

RADIO Frequency Identification (RFID) tags for the function of next-generation electronic product code (EPC) will become one of the most widely used devices in the near future [1]. An RFID application contains three basic roles: 1. tag, 2. reader, and 3. back-end database. Each tag contains a unique identification, often called the tag identification (TID). The reader is used to query the tag's TID and forward it to the back-end database. Once the tag is found valid, the back-end database will look up its product information for further processing. RFID tags are classified into three types: active, semi passive, and passive. Active tags contain batteries so that they can actively communicate with the reader. Semi passive tags also contain batteries but they wait for the reader's query. As for passive tags, the power comes from the reader. The class of a tag represents the effective reading range. The reading range of a class-0 tag is 5-10 cm, and that of a class-1 tag is up to several meters. EPCglobal class-1 generation-2 (Gen2 in brief) was approved as ISO18000-6C in July 2006. It is widely believed that Gen2 tags will be the mainstream when developing RFID applications because the effective reading range is larger. However, the Gen2 specification has the vulnerability that the TID is transmitted without any guard. Thus in this paper, we focus on the protection of class-1 passive Tags from being accessed by malicious readers.

II. SYSTEM ANALYSIS

Existing System

Radio Frequency Identification (RFID) systems are a common and useful tool in manufacturing, supply chain management and retail inventory control. Optical barcodes, another common automatic identification system, have been a familiar packaging feature on consumer items for years. Due to advances in silicon manufacturing technology, RFID costs have dropped significantly. In the near future, low-cost RFID "electronic product codes" or "smart-labels" may be a practical replacement for optical barcodes on consumer items. Unfortunately, the universal deployment of RFID devices in consumer items may expose new security and privacy risks not present in closed manufacturing environments. This presents an introduction to RFID technology, identifies several potential threats to security and privacy, and offers several practical proposals for efficient security mechanisms. We offer several policy suggestions and discuss various open questions and areas of research.

Proposed System

In this project, we propose a novel authentication protocol based on Gen2, called Gen2+, for low-cost RFID tags. Our protocol follows every message flow in Gen2 to provide backward compatibility. Gen2+ is a multiple round protocol using shared pseudonyms and Cyclic Redundancy Check (CRC) to achieve reader-to-tag authentication. Conversely, Gen2+ uses the memory read command defined in Gen2 to achieve tag-to-reader authentication. We show that Gen2+ is more secure under tracing and cloning attacks.

2.1 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ♦ ECONOMICAL FEASIBILITY
- ♦ TECHNICAL FEASIBILITY
- ♦ SOCIAL FEASIBILITY

Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

1) Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

Hardware Requirements:

•	System	: Pentium Iv 2.4 Ghz
•	Hard Disk	: 40 Gb
•	Floppy Drive	: 1.44 Mb
•	Monitor	: 15 Vga Colour
•	Mouse	: Logitech.
•	Ram	: 256 Mb

Software Requirements:

•	Operating system	:- Windows XP Professional
•	Front End	: - Asp .Net 2.0.
•	Coding Language	:- Visual C# .Net
•	Back-End	: - Sql Server 2000.

III. SOFTWARE ENVIRONMENT

3.1 Features of Net

Microsoft .NET is a set of Microsoft software technologies for rapidly building and integrating XML Web services, Microsoft Windows-based applications, and Web solutions. The .NET Framework is a language-neutral platform for writing programs that can easily and securely interoperate. There’s no language barrier with .NET: there are numerous languages available to the developer including Managed C++, C#, Visual Basic and Java Script. The .NET framework provides the foundation for components to interact seamlessly, whether locally or remotely on different platforms. It standardizes common data types and communications protocols so that components created in different languages can easily interoperate.

“.NET” is also the collective name given to various software components built upon the .NET platform. These will be both products (Visual Studio.NET and Windows.NET Server, for instance) and services (like Passport, .NET My Services, and so on).

The .Net Framework

The .NET Framework has two main parts:
1. The Common Language Runtime (CLR).
2. A hierarchical set of class libraries.

The CLR is described as the “execution engine” of .NET. It provides the environment within which programs run. The most important features are

- ♦ Conversion from a low-level assembler-style language, called Intermediate Language (IL), into code native to the platform being executed on.
- ♦ Memory management, notably including garbage collection.
- ♦ Checking and enforcing security restrictions on the running code.
- ♦ Loading and executing programs, with version control and other such features.
- ♦ The following features of the .NET framework are also worth description:

Managed Code

The code that targets .NET, and which contains certain extra Information - “metadata” - to describe itself. Whilst both managed and unmanaged code can run in the runtime, only managed code contains the information that allows the CLR to guarantee, for instance, safe execution and interoperability.

Managed Data

With Managed Code comes Managed Data. CLR provides memory allocation and Deal location facilities, and garbage collection. Some .NET languages use Managed Data by default, such as C#, Visual Basic.NET and JScript.NET, whereas others, namely C++, do not. Targeting CLR can, depending on the language you’re using, impose certain constraints on the features available. As with managed and unmanaged code, one can have both managed and unmanaged data in .NET applications - data that doesn’t get garbage collected but instead is looked after by unmanaged code.

Common Type System

The CLR uses something called the Common Type System (CTS) to strictly enforce type-safety. This ensures that all classes are compatible with each other, by describing types in a common way. CTS define how types work within the runtime, which enables types in one language to interoperate with types in another language, including cross-language exception handling. As well as ensuring that types are only used in appropriate ways, the runtime also ensures that code doesn’t attempt to access memory that hasn’t been allocated to it.

Common Language Specification

The CLR provides built-in support for language interoperability. To ensure that you can develop managed code that can be fully used by developers using any programming language, a set of language features and rules for using them called the Common Language Specification (CLS) has been defined. Components that follow these rules and expose only CLS features are considered CLS-compliant.

The Class Library

.NET provides a single-rooted hierarchy of classes, containing over 7000 types. The root of the namespace is called System; this contains basic types like Byte, Double, Boolean, and String, as well as Object. All objects derive from System. Object. As well as objects, there are value types. Value types can be allocated on the stack, which can provide useful flexibility. There are also efficient means of converting value types to object types if and when necessary.

The set of classes is pretty comprehensive, providing collections, file, screen, and network I/O, threading, and so on, as well as XML and database connectivity.
The class library is subdivided into a number of sets (or namespaces), each providing distinct areas of functionality, with dependencies between the namespaces kept to a minimum.

Languages Supported By .Net

The multi-language capability of the .NET Framework and Visual Studio .NET enables developers to use their existing programming skills to build all types of applications and XML Web services. The .NET framework supports new versions of Microsoft’s old favorites Visual Basic and C++ (as VB.NET and Managed C++), but there are also a number of new additions to the family.

Visual Basic .NET has been updated to include many new and improved language features that make it a powerful object-oriented programming language. These features include inheritance, interfaces, and overloading, among others. Visual Basic also now supports structured exception handling, custom attributes and also supports multi-threading.

Visual Basic .NET is also CLS compliant, which means that any CLS-compliant language can use the classes, objects, and components you create in Visual Basic .NET.

Managed Extensions for C++ and attributed programming are just some of the enhancements made to the C++ language. Managed Extensions simplify the task of migrating existing C++ applications to the new .NET Framework.

C# is Microsoft's new language. It's a C-style language that is essentially "C++ for Rapid Application Development". Unlike other languages, its specification is just the grammar of the language. It has no standard library of its own, and instead has been designed with the intention of using the .NET libraries as its own.

Microsoft Visual J# .NET provides the easiest transition for Java-language developers into the world of XML Web Services and dramatically improves the interoperability of Java-language programs with existing software written in a variety of other programming languages.

Active State has created Visual Perl and Visual Python, which enable .NET-aware applications to be built in either Perl or Python. Both products can be integrated into the Visual Studio .NET environment. Visual Perl includes support for Active State's Perl Dev Kit.

IV. IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods