Addis Ababa University

College of Natural Sciences

RFID Security through Dynamic tag content Management

Bemenet Kasahun

A Thesis Submitted to the Department of Computer Science in Partial Fulfillment for the Degree of Master of Science in Computer Science.

Addis Ababa, Ethiopia

February, 2020

Addis Ababa University

College of Natural Sciences

Bemenet Kasahun Gebremeskel

Advisor: *Dagmawi Lemma (PhD)*

This is to certify that the thesis prepared by *Bemenet Kasahun*, titled: *RFID Security through Dynamic Tag Content Management* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

Name	Signature	Date
Advisor: Dagmawi Lemma (PhD)	_____	_____
Examiner:_____	_____	_____
Examiner:_____	_____	_____

Abstract

Radio frequency identification (RFID) tag exchanges data with an RFID reader through radio waves. These tags can be attached to almost any object, such as baggage's, containers, construction materials, laundry and bottles. It can also be attached to animals, humans and vehicles. It is seen as a means to enhance efficiency and introduce new functionality in products such as intelligent fridges or washing machines, to query their contents. However, concern has arisen about the possibility of using RFID technology for tracking and profiling individual people. Privacy, integrity, data security and civil rights concerns are expressed and may lead to the failure of RFID technology to realize its promise.

Cryptographic solutions may be a consideration. However, standard cryptographic primitives (hash functions, message authentication codes, block/stream ciphers, etc.) are quite demanding in terms of circuit size, power consumption and memory size, so they make costly solutions for RFID tags. We analyze the security hole and present these concerns and find out the problem as static tag_id identification and no access control. As we see none of RFID tags employ read passwords or other read access control. Because the tag content on the RFID tag never changes until next encoding, the ability to read will enable several security risks. First, the adversary may determine which object owns the tag and infer the origin of the object carrying the tag. Second, static identifier can be used both to track and hotlist tagged objects easily. The main security risk of the static tag content is the leakage of content and cloning the tag. Because of the static properties, RFID tag content always remains unchanged in the entire system, and an attacker can conducts multiple actions on the target tag. Tag that stores static information is vulnerable to attacks such as cloning. So our proposed solution comes with the idea of using dynamic tag content to the tag which is making the tag_id dynamic by changing it every time the tag is read/scanned. The tags store a dynamic number, called tag_id. The back-end system issues these numbers and keeps track of which number is written on which tag to prevent cloning attack. This work makes use of the tag's rewritable memory for changing tag content after every scan. Finally, we simulate our work using Arduino and Proteus simulation tools.

Keywords: RFID, Privacy, integrity, data security, Cryptography, authentication, cloning.

Dedicated to:

1. My advisor Dagmawi Lemma (PhD)
2. My father Kassahun Gebremeskel who was unexpectedly died during the final stage of this thesis. R.I.P.

Acknowledgments

Thanks to God for giving me the power and courage to complete my thesis successfully. This paper appears in its current form due to the assistance and guidance of several people. Therefore I would like to express my sincere thanks to all of them. My special thanks go to my Advisor Dr. Dagmawi Lemma for his excellent guidance and support throughout the thesis.

To all my friends and family, especially to my brother (Shemeles kassahun) and my sister (Yewubdar Kassahun), they provided me a great support throughout the various stages of the work. Last but not the least, I would like to say thank you my dear mother Mulu Gemmach

Table of Contents

List of Figures

List of Tables

List of Algorithms

Acronyms and Abbreviations

AIDC	Automatic Identification and Data Capture
ASIC	Application Specific Integrated Circuit chip
Auto-ID	Automatic Identification
DOS	Denial of service
EEPROM	Electrically Erasable Programmable Read Only Memo
HF	High Frequency
IC	Integrated Circuit
LF	Low Frequency
OCR	Optical Character Recognition
RF	Radio frequency
RFID	Radio Frequency Identification
SPI	Serial Peripheral Interface
UbiComp	Ubiquitous Computing
UHF	Ultra-High Frequency

CHAPTER 1: INTRODUCTION

1.1 Background

Radio frequency identification (RFID) is a fast developing technology that provides wireless identification and tracking capability. It helps us on many applications such as preventing theft of automobiles and merchandise, collecting tolls without stopping, gaining entrance to buildings, controlling access of vehicles to gated communities, corporate campuses and airports, tracking library goods, asset identification, retailing and supply chain management, animal tracking, among others, take advantage of RFID systems [1, 2]. RFID is generally characterized by the use of simple devices on one end, called tags or transponders, and more complex devices on the other end of the link, called readers or interrogators. The tags are made up of an antenna and an Application Specific Integrated Circuit (ASIC) chip, which contains memory where data is stored. Occasionally, they can include a matching network, located in between the antenna and the chip, to achieve proper impedance matching. The readers are composed of an antenna, an RF(Radio frequency) electronic module, which is responsible for communicating with the tag, and a control electronic module, which is responsible for communicating with a host computer (or controller), usually connected to the reader in order to centrally process information coming from readers. RFID technology is a prominent area of research in ubiComp. Its contactless nature and potential for data processing and storage gives it many advantages over existing machine readable identification techniques (e.g., barcodes, optical character recognition) [5].

1.1.1 Automatic Identification Systems

Auto-ID (Automatic Identification) technology, also called Automatic Identification and Data Capture (AIDC), is a big set of identification procedures which include the very well-known barcode, as well as optical character recognition (OCR), infrared identification and RFID. Among the various forms of Auto-ID traditional one-dimensional barcode (UPC code) Figure 1.1 and, two dimensional barcode in the Figure 1.2 (QR code) dominate the Auto-ID market being used in almost everything and everywhere today in the world.

Figure 1.1: Traditional one dimension barcode

Figure 1.2: Two dimensional barcode

The main reason is their ultra-low cost, which is almost negligible. However, they are limited in memory storage capability and 'line of sight' operation is required. The latter makes the presence of an operator necessary to read a barcode. Since RFID systems do not have these limitations and, therefore, remove the human intervention in the reading process, RFID technology is coming into the Auto-ID market with a huge potential. In addition, unlike barcodes, RFID technology provides security by means of data encryption, and read/write capability. Nevertheless, RFID tags require a chip to store the data, which makes the tags expensive to be implemented in certain Auto-ID market applications. Barcodes typically cost under 0.01 euros, whereas RFID tags cost over 0.10 euros [2, 3]. Thus, there is a big demand of low-cost RFID technology around the world, and it is believed that someday RFID tags will be as pervasive as barcodes [25, 26].

1.1.2 Operating Frequency Ranges and Applications

RFID systems operate at widely different frequency bands. The main frequency bands used in RFID technology is discussed below [2, 3].

- **Low frequency (LF)**. These systems operate between 125 KHz and 134.2 KHz. Due to the electromagnetic properties at these frequencies, LF tags can be read even when they are attached to objects containing water, animal tissues, metal, wood, and liquids. However, they are only suitable for proximity applications, because they can be interrogated from a very short range of only a few centimeters (generally LF tags are passive). LF tag antennas are usually made of a copper coil with hundreds of turns rolled around a ferrite core. Because of these properties of LF tags, they are used for specific applications such as animal identification, access control, asset tracking, vehicle immobilizer, healthcare, and various points of sale applications. In particular, LF tags have been intensively used for animal tracking since the early 1980s.

- **High frequency (HF).** Working around a central frequency of 13.56 MHz, HF tags are passive and their operating principles are similar to LF tags. However, HF tags have a better read range than LF tags and can be read up to half a meter away. The tag includes an LC resonant antenna usually made of several turns (~ 5-20 turns) of conductive materials such as copper, aluminum, or silver as a flat.

1.2 Motivation

Technology like sensors, actuators, RFID tags assemble themselves into our daily life until they are indistinguishable from the smart environment. In creating such smart environment security is one issue. Thus, analysis of security issues that lead to vulnerability of the system is prevalent in the design and development of RFID systems which is one component of ubiComp. Hence, we are motivated to ensure that information is not stolen, modified, forged or accessed illegally. As long as RFID tags do not comply with open and community reviewed encryption standards the security of these tags need to be independently assessed.

1.3 Statement of the Problem

RFID is considered to be one of the important technological building blocks for ubiquitous computing. RFID tags are also being embedded in everyday objects and accessed by a networked reader's infrastructure [10]. These tags are small, wireless devices to help identify objects as well as people. RFID systems are used to track objects in supply chains, and are working their way into the pockets, belongings, and even the bodies of consumers. While widely being used, RFID technology has security vulnerability since RFID tags are considered "dumb" components in that they can only listen and respond, no matter who sends the signal requesting the data inside the tag. Attack points of RFID system is illustrated in the Figure 0.3 [14].

This brings risks like unauthorized access to the tag. Unprotected tags can be vulnerable to eaves dropping, tag content changes, man in the middle attacks, tag cloning, relay attacks and unauthorized tag reading [34].

- **Eavesdropping**

  In eavesdropping, hackers secretly monitor data sent from an RFID tag to a reader, or vice versa, via the air interface (the communication channel between the reader and tag). Because eavesdropping is passive, attackers do not emit any signal and it is highly difficult to detect.

- **Tag content changes**

  If a tag is writeable, attackers can change its content, distorting item attributes or leading the access control system to falsely reject an authorized person. Furthermore, they can insert malware such as modified tag data that the reader interprets as a command.

- **Tag cloning**

  In tag cloning, attackers make a duplicated RFID tag, which might either be quite similar in size or much larger than the original but have the same functionality. Attackers can use duplicates to access a restricted area, abuse private data, or make an electronic transaction on the victim's behalf.

- **Man-in-the-Middle Attacks**

  Related to the relay attack is the Man-in-the-middle attack, which is a type of attack where attackers intrude into a communication channel to intercept the communications and possibly inject data into the communications. A Man-in-the-middle attack involves a situation where the attacker creates independent connections with the interrogator and the RFID tag by relaying messages between them.

- **Relay attacks**

  In a relay attack, attackers create a connection between a legitimate reader and a victim's legitimate tags. From the RFID system's view point, the communication looks as if the legitimate tag and the reader are close to each other, when in fact they are communicating through the communication channel that the attackers have established.

- **Unauthorized tag reading**

  Attackers can use a fake reader to read tag information. They can extend fake reader's range by several times that of the standard communication distance.

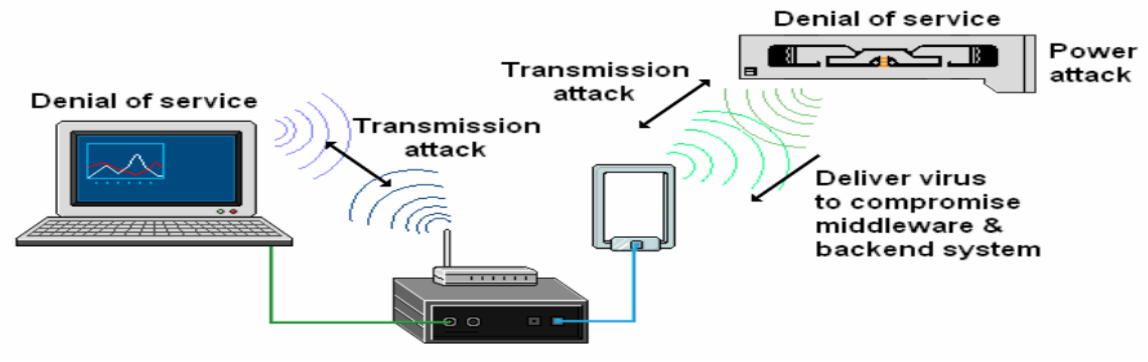


Figure 1.3: Attack points

Some works tried to solve existing security problems from various perspectives**,** but have the following gaps.

1. Capacity analysis is not done as if it's hard to implement cryptographic solution for RFID tag.
2. Previous works did not consider tag content with regard to tag data modification and cloning the tag.
3. However, tag content is encrypted but still it can be cloned.

Thus, this research work is meant to address one of the above mentioned problems which is tag cloning attack.

1.4 Objectives

General Objective

The general objective of this research is to solve RFID tag cloning attack.

Specific Objectives

To achieve the general objective, the following specific objectives will be done.

- Review researches that have been done in the area.
- Assess and select different security mechanisms which are relevant for RFID technology.
- Propose and develop appropriate security mechanisms for RFID tag technology.
- Develop Dynamic tag content management.
- Simulate the work.

1.5 Methods

In order to accomplish the objectives of the research we apply the following methods and procedures.

Literature Review

Reviews of the relevant literature will be done to have a better and solid understanding of the research area. A number of published research papers, thesis works, books and web sites in the area of RFID technology security and privacy will be studied.

Design Science

We develop artifact of the proposed solution.

Simulation

Simulation tools will be selected to simulate the proposed work.

1.6 Scope and Limitations

Scope

The scope of this thesis is to develop RFID security through dynamic tag content management.

Limitations

We can't simulate full of the work because Proteus simulation tool lacks RFID RC522 (the module we use) read/write module, and also we need the hardware components integrated with the software for implementing full proposed system. In addition, the reader should incorporate encoding capability in addition to reading the tag content.

1.7 Application of Results

The outcome of this research primarily can be used by RFID system designers. Designers can follow this work in designing RFID technology for improving security of the tag; in particular improve security of RFID based applications.

1.8 Organization of the Rest of the Thesis

Organization of the Rest of the Thesis: Chapter two presents the literature review. Chapter three discuss the related work which had been done in the area of the problem and those related to our work. Chapter Four details the proposed design of this thesis. Implementation and simulation of the proposed solution is discussed in Chapter Five. And Chapter Six deals with the recommendation and conclusion of this thesis work.

CHAPTER 2: LITERATURE REVIEW

2.1 RFID Technology

Using RFID Technology it is possible to identify objects without line of sight or touching them, it is possible to track items or to trigger actions through implicit interaction. The tag is usually attached to an object that is to be identified. Radio transmissions are used by the reader to send a query to the tag and by the tag to return an answer, generally containing identifying information. The reader also can be connected to a host computer, where information can be incorporated into a database [33].

TID based tag verification scheme confirms that TID numbers currently provide a practical difficulty against cloning of Gen-2 chips since Gen-2 chips with programmable TID memory. However, working prototypes of semi passive tags demonstrate that a tag impersonation device can be built from less than ten euros worth of standard components to fool TID checks. As a result, end-users should only make use of TID numbers in applications where the tagged items can be physically inspected as a temporal and complementary solution. So when end-users completely rely on TID checks he/she could create a rewarding opportunity for manufacturing programmable chips that would completely undermine the practical hurdle that the TID scheme provides today. Overall, the biggest threat against this scheme relates to the commoditization of RFID technology. Therefore, TID numbers do not appear to provide any sustainable long term solution for tag cloning, but only a temporary solution before stronger tag authentication techniques [35].

Mostafa and Ira developed a system to implicitly enforce RFID tag authentication using kill passwords [30]. There were complaints from the public that large commercial outfits like Wal-Mart and Shoprite can track customers with the tags attached to goods [13]. So the idea to kill tags after purchasing products that have RFID tags was born. According to electronic product code global (EPC Global), which is the universal RFID standard, killing a tag permanently disables the tag [4]. However, when a reader issues the kill command, it only zeroes out a tag's memory bank [3]. Furthermore, it is possible for programmers to reprogram the tag and bring it back to life. Lee, and J. Kim [6], saw it as a possibility of using this feature to prevent and detect tag cloning in low-cost RFID tags. During the process of killing a tag, any error detected will either take the tag back to an arbitrary state

or keep the tag in its current state. If no errors are identified during this process, the tag moves on to be killed.

The EPC Global standard implied that when an RFID tag receives a correct password with a power level that is insufficient to kill a tag, it replies with an error code [12]. If a tag sends a wrong password through the kill command operation, the reader will not respond at all. So tag must be programmed with the correct password and just about the right amount of power to insufficiently kill the tag. Mustafa and Ira [30], described that for tag authentication to take place instead of the actual killing of the tag, the tag needs to request the appropriate amount of power from the reader and different tags require different power levels to effectively carry out this authentication.

The synchronized secrets prototype demonstrates that raises an alarm as soon as two tags with the same ID are scanned within a supply chain. The additional cost factor of the presented method is manual verifications needed to determine which of the tags (objects) with the same id number is the cloned one, but the number of needed verifications for the presented method is considerably smaller than for comparable detective security measures. Overall, the presented method has the potential to make harmful injection of cloned tags into RFID systems [7].

In Kill password method, the communication between the tag and reader occur at a faster rate compared to the synchronized secret. The reader has to check for the accuracy of the password from the backend only once in a single authentication process. After that, the backend is no longer directly involved in the tag authentication and kill command process [6]. This results in a relative increase in authentication rate. One of the hardware constraints of a low-cost passive tag is that it does not have a user defined memory space. So for a system to use the synchronized secret methodology in a cheap tag, one has to overwrite the 4bytes reserved for the access password with a synchronized secret [7]. The access password can be completely or partly overwritten, depending on the secret, the more secure the system, but the more time it will take to transmit. A decision has to be made based on the area of use and the administrator's preferences. In other words, synchronized secret requires a certain level of configuration to function properly.

Kill passwords also require a degree of configuration. As a security measure, an administrator might decide to change the default passwords on the tags. Overwriting is not needed since the technique makes use of the tag's access password. The main configuration in kill passwords takes place when selecting the appropriate power level to activate the kill command. It was inferred that different tags require different power levels to respond with the "insufficient to kill" message [6]. Therefore the administrator needs to know the amount of decibels (dB) to configure on the tag. The configuration will enable the reader know how much power to emit to produce the required response from the tag.

In terms of downright security, it can be argued that the kill password is more secure to tag cloning attacks. This is because the kill command technique carries out authentication based on three criteria. These are the TID, the password and the power level. In comparison, synchronized secrets make use of only the TID and a secret. The extra layer of security in the kill technique suggests that it can be more resistant to cloning attacks. However, an extra layer of security does not necessarily guarantee more security. If weak passwords are used an RFID system that uses kill passwords and the attacker is able to determine the required power level, the added security layer might not be able to detect or prevent the attack. Therefore, the kill password largely depends on the strength of the password used and the unpredictability of the required power level.

Symmetric-key cryptography can be used to avoid tag cloning attacks. The tag (Ti) shares the key (Ki) with the reader. Afterwards, the following messages are exchanged.

- The reader generates a fresh random number (R) and transmits it to the tag.
- The tag computes H = g(Ki, R) and sends it back to the reader.
- The reader computes H0 = g(Ki0, R) and checks its equality with H.

The *g* function can be implemented by a hash function or, alternatively, by an encryption function. Note that if the *g* function is well constructed and appropriately deployed, it is infeasible for an attacker to impersonate the tag. But standard cryptographic primitives (hash functions, message authentication codes, block/stream ciphers, etc.) are extravagant solutions for RFID tags on account of their demand for circuit size, power consumption and memory size [9].

Demetrious [11] review the proposed Gen2 security framework and introduce cryptographic suites to illustrate how to utilize this framework to provide a range of security functionality. They analyze the security of the Gen2 protocol and this new functionality in the context of timing based attacks. They conclude that the tight communication timings specified in the Gen2 protocol mitigate timing based attacks however; the loose timing implementations on commercial interrogators and limited timing enforcement on tags lesson the effectiveness of the specified timing constraints. Further, they conclude that the new security framework allows for the efficient integration of secure functionality that, as specified, is resistant to timing based attacks. EPCglobal developed standard security framework within which security functionality integrated seamlessly into the Gen2 protocol. The new Gen2 security framework has six primary commands that may be used with a cryptographic suite: Challenge, Authenticate, ReadBuffer, SecureComm, AuthComm, and KeyUpdate. The primary goal of these commands is to provide a frame work for cryptographic identity authentication (Challenge and Authenticate) and a secured and/or authenticated communication channel (SecureComm and AuthComm). The KeyUpdate command allows for the explicit changing of cryptographic keys. The new security framework allows for a tag to take an extended period of time (several tens of milliseconds or longer) to complete its operations. They refer to a response that is allowed to take more than T1 time as a delayed response. A tag requiring a significant amount of time to complete its operations will beacon a 'busy signal' at least every 20ms to indicate to the interrogator that it is still computing its result. A tag instead of communicating its response directly to the interrogator may write its result to a new response buffer and, after doing so, indicate to the interrogator that it is finished. The interrogator can read the contents of this buffer through the ReadBuffer command. The Challenge command is a broadcast command that contains at least 48 bits that form a packet wrapper around a message that is to be interpreted according to some cryptographic suite. The content of the message is defined by the cryptographic suite being used. The Challenge command contains fields within it that enables the interrogator to specify which cryptographic suite is to be used. Authenticate command is a singulated command that contains at least 64 bits that form a packet wrapper around a message that is to be interpreted according to some cryptographic suite. The content of the message is defined by the cryptographic suite being used. The Authenticate command contains fields

within it that enables the interrogator to specify which cryptographic suite is to be used. Similarly, the AuthComm command is a singulated command that contains at least 42 bits that form a packet wrapper around an authenticated message that is being communicated to the tag. This message will typically be a command, such as a Write command or a Read command that is authenticated according to the cryptographic suite that was established during the authentication process. The Read Buffer command is a singulated command that contains 67 bits and is used to retrieve the contents of the response buffer within the tag. However, using the delayed response of the new Gen2 security functionality creates new vulnerabilities to timing based attacks such as relay attacks and man-in-the-middle attacks. And GEN 2 protocol (publically available) known so the communication between the tag and the reader can easily eavesdrop and also their work is not responsible for securing at singulation stage.

2.2 How RFID Work

RFID belongs to a group of technologies referred to as Automatic Identification and Data Capture (AIDC). AIDC methods automatically identify objects, collect data about them, and enter those data directly into computer systems with little or no human intervention. RFID methods utilize radio waves to accomplish this. At a simple level, RFID systems consist of three components: an RFID tag or smart label, an RFID reader, and an antenna. RFID tags contain an integrated circuit and an antenna, which is used to transmit data to the RFID reader (also called an interrogator). The reader then converts the radio waves to a more usable form of data. Information collected from the tags is then transferred through a communications interface to a host computer system, where the data can be stored in a database and analyzed at a later time. To read the information encoded on a tag, a two way radio transmitter receiver called an interrogator or reader emits a signal to the tag using an antenna. The tag responds with the information written in its memory bank. The interrogator will then transmit the read results to an RFID computer program [27]. Figure 2.1: deal with RFID System structure and interaction of each component [15].

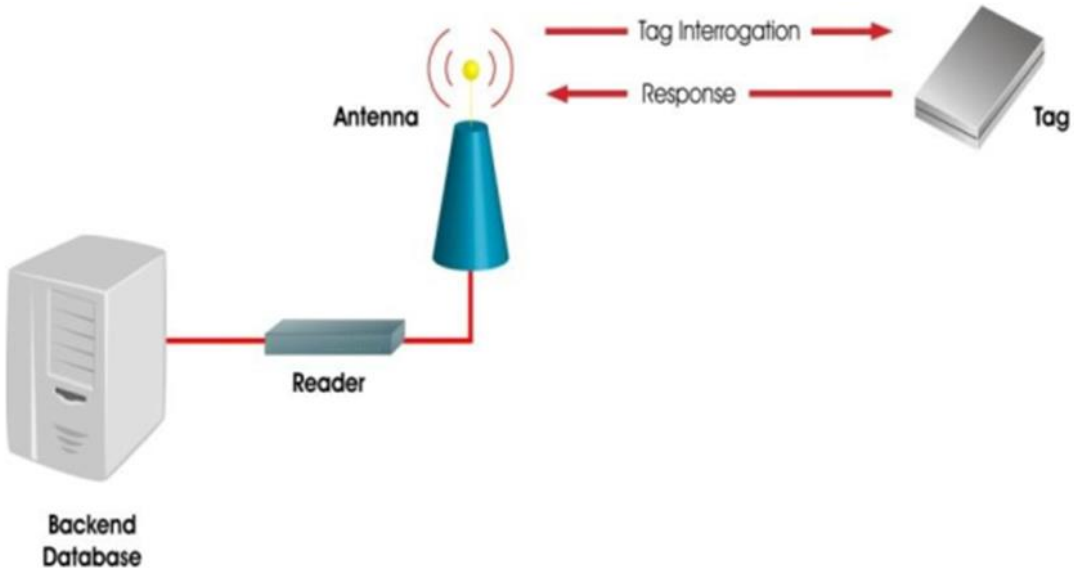


Figure 2.1: RFID System structure

2.3 Types of RFID systems

RFID system can be broken down by the frequency band in which it operates. Whether it is low, high or ultra-high frequency there are three major categories of RFID systems. These are active, passive and, semi-passive and are explained below [28, 29].

a. **Passive Tag – Active Reader**

As Figure 2.2 shows Passive, Passive tags have no power of their own instead; all the power needed to operate the tag is derived from the radio signals sent by the reader. Furthermore, passive tags have no conventional radio transmitter and, as such, cannot create their own signal. Instead, they vary the electrical load attached to the antenna in order to vary the signal reflected from the antenna, somewhat analogous to using a movable mirror to send a signal by reflecting the light of the sun towards a watcher. This technique is known as back scatter communications. Back scatter communications also requires that the tag and reader be in close proximity (within a few feet) of each other when the transmitted radio waves are low frequency (LF) or high frequency (HF). With the emerging technology that transmit ultra-high frequency (UHF) radio waves, longer read ranges are becoming possible1; some UHF RFID manufacturers claim read ranges as long as 15 to 20 feet. Since metal and water (or moisture) tends to absorb UHF waves, one has considered their presence when evaluating UHF tags read ranges. Passive tags often are extremely simple devices. The

typical structure consists of a plastic substrate or inlay, a printed or etched metal antenna, and a single integrated circuit. As a consequence, passive tags can be much smaller and less expensive than other types of radio devices.

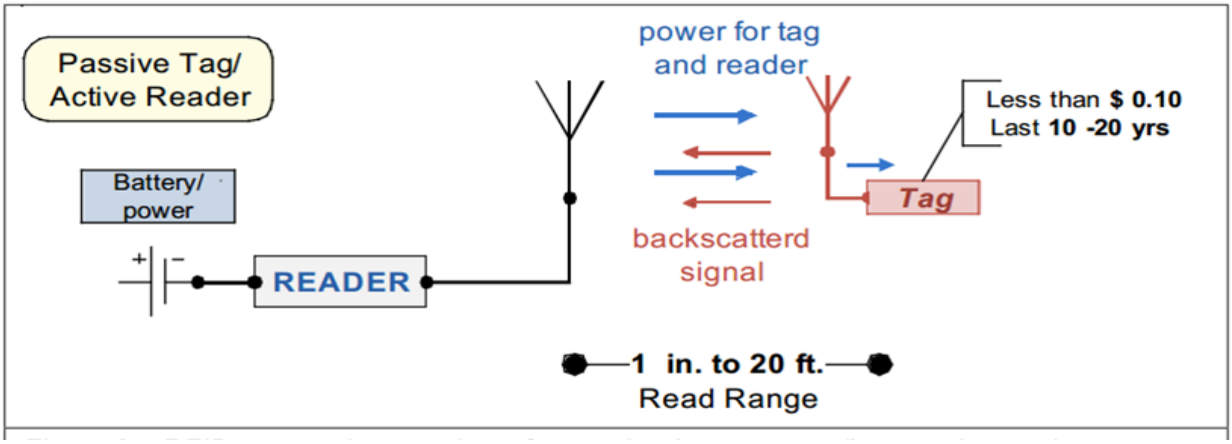


Figure 2.1: RFID systems that consist of passive tag and an active reader

A passive tag requires no maintenance, and has a long lifetime, limited by degradation of the tag materials rather than battery usage. It is reasonable to expect that in many environments passive tags will be readable for 10 to 20 years and it is illustrated in Figure 2.2.

b. **Active Tag – Passive Reader**

In the Active Tag – Passive Reader system, the tag is powered and the reader has no power. In essence, an active RFID tag is equipped with its own radio transmitter, such as a cellphone or Wi-Fi client. Active tags use conventional circuitry for transmission and reception with read range and reliability similar to the performance of other radios. As such, read ranges of hundreds of meters to kilometers (miles) are achievable, and tags can be read despite substantial obstructions between the tag and reader. With improved read range and reliability comes increased cost, size (circuit complexity), and maintenance requirements. Active tags cost more than $20 each; some cost as much as $100 each and are designed for tracking high value assets and it is presented in Figure 2.3.

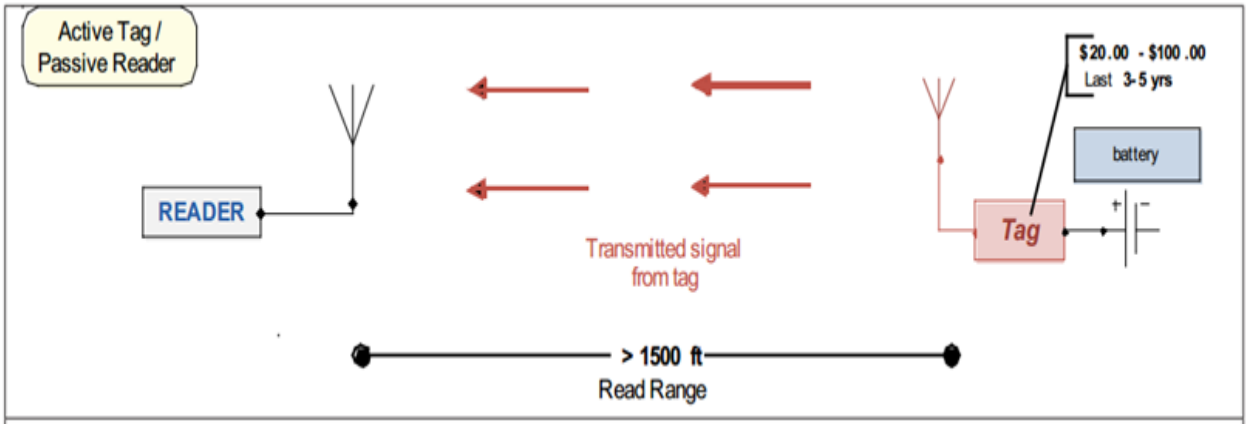


Figure 2.2: RFID system that consists of active tag and passive reader

c. Semi-Passive Tag / Active Reader

This RFID system is intermediate between the two systems discussed in the above sections. In the semi passive tag / active reader system, the reader is powered and the tag has a battery to power the tag's circuitry; the tag, however, still employs backscattering or load modulation to communicate with the reader.

Semi passive tags require that the reader signal be large enough to decipher but does not need to extract power to run the tag's circuitry. Therefore, the read range is not limited to a few feet (as with a passive tag). The read range of semi passive tags usually is limited by the rapid decrease in the reader signal; however, high quality receivers can achieve read ranges on the order of 300 feet in unobstructed areas. In addition to longer range, semi passive tags provide much better reliability at short ranges. Figure 2.4 depicts that passive tag at several feet from a reader might not read the tag if the transmission path (of the radio waves) from tag to reader is blocked by an obstacle. A semi passive tag at a similar distance is often better to decipher and reply to the reader's signal. A downside of semi passive tags is that they are more expensive than passive tags; furthermore, their applicability is limited by battery life. Manufacturers of semi passive tags try to ensure that most of the circuitry is switched off except when the tag is being queried by a reader; battery life is still generally limited to only a few years.

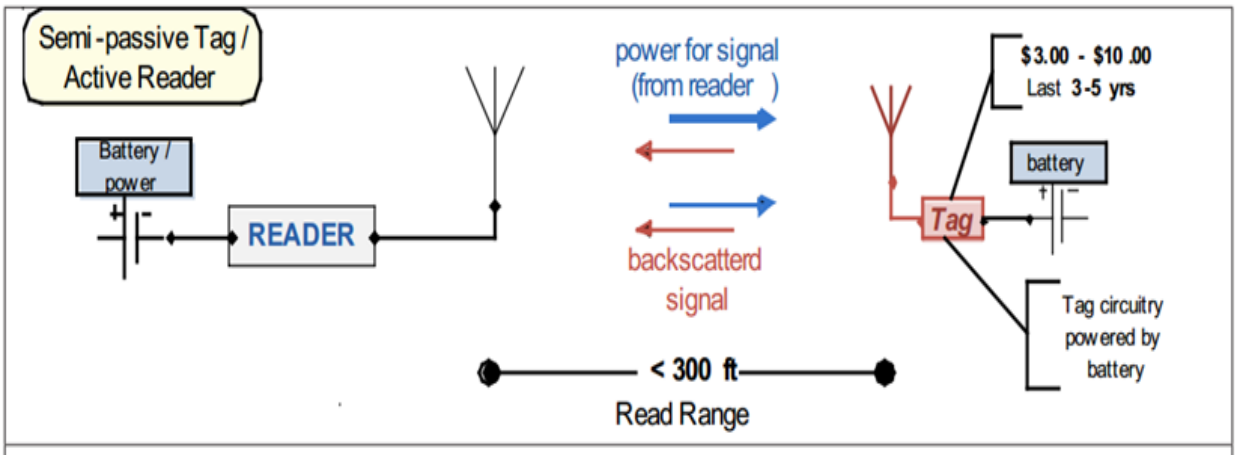


Figure 2.3: RFID system that consists of semi passive tag and active reader

2.4 Comparison between RFID and Barcode Technology

No human capital is required with RFID system and it is completely automated. On the other hand a dedicated employee is required to scan barcodes of items. Barcodes can only be read while RFID can not only be read but also rewritten and modified depending upon requirements.

While barcodes can be easily damaged and are difficult to read when greasy or dirty, RFID is rugged and extremely durable. Barcodes can be counterfeited or reproduced whereas this is not possible in the case of RFID tags. While only one item can be read at a time with a barcode scanner, RFID reader can read up to 40 items per second.

The range of RFID reader is 300 feet. On the other hand barcode scanner can barely read past 15 feet. When comparing RFID and barcode methods, RFID is a more advanced and adaptable technology that will surely be utilized more as cost to implementation decreases [31]. And comparison of RFID tag with Barcode is given in Table 2.1.

Table 2.1: Comparison Summary

Parameters	**RFID**	**Barcode**
Line of sight	Not required	Required
Read Rate	10 100 or 1000 simultaneously	Only one at a time
Identification	Can uniquely identify each item/asset/object tagged	Most barcodes only identify the type of item (UPC code) but not uniquely
Read/write	Many RFID tags are Read/write	Read only
Technology	RF	Optical (laser)
Interference	Like the TSA (Transport Security Administration), some RFID frequencies don't like Metal and Liquids. They can interfere with some RF frequencies.	Obstructed barcodes cannot be read (dirty covering barcode, torn barcodes, etc.)
Automation	Most fixed readers don't require human involvement to collect data (automated)	Most barcode scanners require a human to operate (labor intensive)

2.5 Applications of RFID

The RFID technology has become wide spread due to its low cost and easy deployment. This section briefly introduces several applications using RFID technology that are widely deployed in our daily life. The widest application of RFID is supply chain and logistics [15]. Products attached with RFID tags can be monitored in the whole supply chain. For each RFID supply chain procedure, numerous RFID tags are functional as the quantity of goods that are involved. Thus, passive tags with low price and recyclability will reduce the overall cost and improve the efficiency of inventory tracking and management.

Access control is another typical application of RFID tags [16]. They can be used as an access key to pass the security entrance of confidential department [17]. They can also be used as a security device to automatically identify vehicles. An RFID tag is attached to a car key so that the automobile can be turn on when the key is close enough, i.e., the reader receives a responding signal from the tag. This application is reported to effectively reduce the auto theft [18].

Animal tracking is one of the oldest applications of RFID technology. Animals have been implanted with RFID tags to help tracking and managing a scientific research. Lost animals can be easily found and returned to their owners by tracking the tags on them [19]. Moreover, scientists have used RFID based animal tracking to observe and control the outburst of animal diseases such as mad cow disease and bird flu.

RFID technology is adopted in the passports of some countries [5]. RFID tag embedded in a passport records the information of the holder. These RFID enabled passports are difficult to forge in comparison with the traditional passports. Therefore, RFID technology can expedite exit and entry for militaries and improve national security.

It has been realized that the traditional payment methods with credit card or cash are quite inefficient. Paying by cards requires customers to sign a receipt to reentering a personal identification number to confirm the payment, while paying by cash needs shop assistants to collect the money and give the change. Recently, some credit card companies began to offer a contactless payment system with RFID technology integrated [18]. The point of sale terminals in this kind of RFID enabled systems can make the checkout procedure more efficient and convenient.

RFID can be embedded into smart appliances. For example, a washing machine with an RFID reader can read the tag of clothes and then run a specific washing process [21]. Smart oven is also capable of reading the instruction from the tag and decide how to cook the food [19].

RFID technology facilitates health care as well [20]. It can be used to efficiently search treatment record for a patient, monitor patient's drug treatment and locate patients. The RFID system also can provide automated processes to reduce the high cost of hospitals and reduce mistakes, as a result it improve safety for patients.

2.6 RFID tag cloning attack

RFID technology operates under the assumption that every tag has a unique identification number. That is, there is no tag with the same identification number. This assumption allows a reader to uniquely identify a tag and know exactly which object is in range of the reader. However, one of the most prevalent security issues in RFID system is the cloning attack on tags and it has been demonstrated by a number of researchers that in fact the uniqueness of tags cannot be guaranteed because their identification numbers can be cloned (copied). In essence, tag cloning is the theft of a tag's identity. By simply having a clone in the system, an attacker can fool the system into believing that it is really identifying the original tag and object [24]. The attacker can then commit RFID enabled crimes using the cloned tag, like: gaining entrance to facilities that are protected by proximity cards; making payments using speed pass tags, and even crimes typically associated with mainstream identity crime such as financial gain, people smuggling, drug trafficking, terrorism and money laundering. In its most elementary form, an attacker only needs to know the identification number of a legitimate tag to clone it. Tag cloning is the most serious theft of tag identity threat, although there are several other potential threats that can be carried out by simply obtaining the identity of a tag, as illustrated in Figure 2.6 [24].

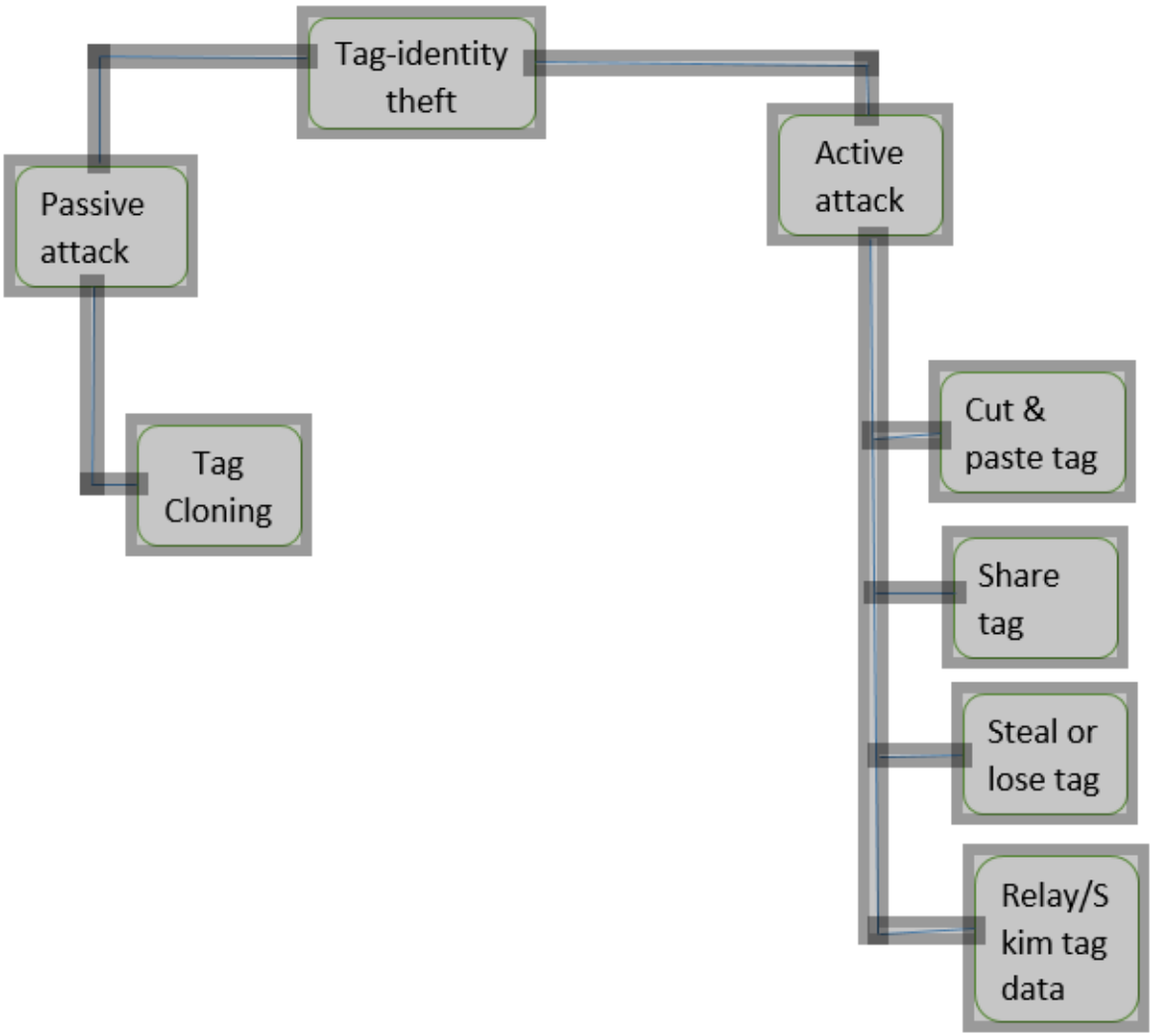


Figure 2.4: Tag identity threat

a. Passive attacks: Tag cloning

Passive attacks involve an attacker copying and storing a tag's identity, and then replaying it at a later time. These types of attacks have greater scope than active attacks because they can be performed repeatedly over time. Tag cloning is one known example of a passive attack. In tag cloning, there are three ways in which an attacker can obtain the identity of a tag: force the tag into revealing it; intercept the communication channel between a tag and a reader; or simply guess a tags identification number [24].

b. Active attack

Active attacks involve the real-time theft and replay of tag identity. Such attacks are limited in scope but have the same overall impact as most identity attacks. These attacks include when a tag is removed from an object and then attached to another object. Users may share their tags with other users or tags may be lost or stolen. These attacks highlight the problem of assuming that a tag is permanently attached to a particular object [29].

2.7 Data security

Data security is the way to ensure that data is kept secure from theft and that access to it can be controlled. Thus, data security helps in protecting personal data. Data security has several aspects [12].

a. Confidentiality

Access must be restricted to those authorized to view the data in question. It is common, as well, for data to be categorized according to the amount and type of damage that could be done should it fall into unintended hands.

b. Integrity

Integrity is the information that should not be altered without the permission of the owner of the information. Common form of attack is the presence of viruses, Trojan horses, or other users change the information without permission, "the man in the middle attack" in which one puts oneself in the middle of the conversation and posing as someone else.

c. Authentication

Authentication is defined as a method to certify that truly original, or people who has access or disclosure of data for legitimate object is needed.

d. Availability

Availability relates to the availability of information when needed. Some of the barriers are found, such as "denial of service attack" (DOS attack), where the server sends requests (usually false) barrage or unexpected demand and therefore cannot serve other requests or even down, hangs, crashes. The barriers also can be mail bomb, where a user sends an email

barrage (thousands of email) with a large size so that the user can not open emails or difficulty accessing email. Information systems are being attacked or hacked to inhibit or eliminate access to information.

e. Access Control

Access control is a way to control access to information which is dealing with the problem of authentication and authorization. The method used is to use a combination of user ID or password or using other mechanisms.

f. Nonrepudiation

Nonrepudiation is an aspect that a person cannot be denied having a deal which supports for electronic commerce.

2.7.1 Cryptography

Cryptography is derived from the Greek "cryptos" means “secret" and "graphein" meaning "writing". Thus, cryptography means "secret writing"(hieroglyph). During the war II, cryptography was used to change the message from the language that is understood to be difficult to understand or even has no meaning. But at the present time, cryptography is not only related to one's privacy, but it also has other functions as data integrity, authentication and nonrepudiation [13].

Cryptography is the art and science of maintaining the security of a message. A word contained in the definition of art is actually derived from the facts of previous early cryptography, where each person has a way to write secret messages. In its development, cryptography can also be viewed as one discipline. Because the techniques that are used in cryptography is mathematically formulated so that it become an official method [14]. Cryptography consists of two main processes. Encrypting the plaintext (original message understood its contents or meaning) into cipher text (the message that is not understood, it is result of the transformation of the plaintext). And this process is called encryption or enciphering. And returning the cipher text into plaintext is called decryption or deciphering [13]. Plaintext can be encrypted and decrypted using a special algorithm called the cipher and a key. Cipher itself actually is a mathematical function while the key is a series of bits that are used to encrypt and decrypt data. The key can be any value from a number of points.

Thus, the level of security is determined by the “key” that means how private the key is, not by the details of the algorithm itself.

CHAPTER 3: RELATED WORK

In this chapter, we review papers that are particularly related to our work. Hence, the various techniques that are related to the RFID system security are reviewed.

3.1 Synchronized Secrets Approach

Lehtonen et al. [22] proposed Synchronized Secrets, in the proposed method the tags store a random number that is changed every time the tag is read. They denote this number a synchronized secret since it is unknown to all who do not have access to the tag and it can also be understood as a one-time password. Every time a tag is read, the back-end first verifies the tag's static identifier. If this number is valid, the back-end then compares the tag's synchronized secret to the one stored for that particular tag. If these numbers match, the tag passes the check otherwise an alarm is triggered. After the check, the back-end generates a new synchronized secret that the reader device writes on the tag Illustration of the method is given in Figure 3.1.

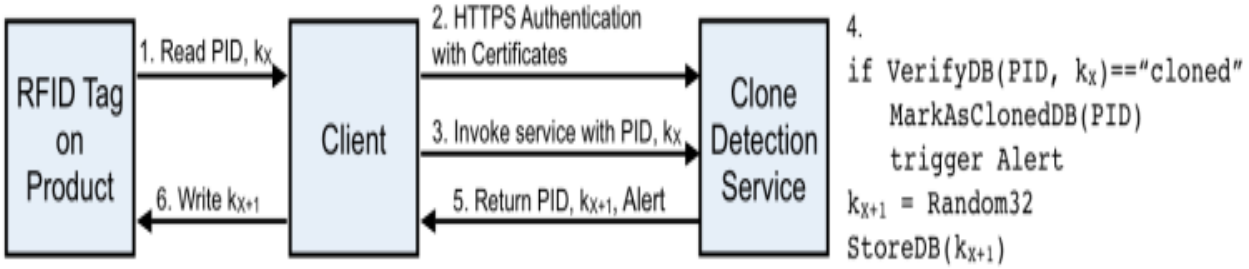


Figure 3.1: Illustration of synchronized secret method

Juels and Weis, [9] proposed security framework in security and privacy problems of RFID system. Their work draws agent and trusted computing technology into the design of RFID system, whose superiority in authentication is used to solve the security problem. This RFID security framework contains three components: some RFID tags, trusted RFID readers that remotely stores and retrieves data using RFID tags or transponders, and a trusted server which contains privacy and security information. Addition to these, there is a database that stores product information, tracking logs, or key management of data. The trusted server

consists of trust policy engine, trust agent pool ANS (agent name service) RFID access module and security module. The TS (Trusted Server) makes RFID network related data available. Trusted Agent Pool is designed to manage trusted agents, including assembling agents, management agent life cycle etc. They introduce a special piece of software, a Trusted Agent (TA) that enables automatic monitoring of read and secure policies. And design two types of agents. One is called Trusted Read Agent (TRA) another is called Trusted Update Agent (TUA). The first one is responsible for transporting read policy that determines which tags an RFID reader is permitted to scan and the permitted uses of the resulting data. The trusted update agent is responsible for updating tag reader privacy and security policies. ANS (Agent name service) provides a global lookup services to translate a unique agent ID into one or more RFID network uniform reference locators where further information on the specific reader may be found.

Trust Policy Engine will provide privacy policies and security policies to the TR (Trusted reader). They set interface in the trust policy engine to allow for the trusted policy to be updated or modified. Building a fully general policy base is likely to be difficult. It's a gradual process. RFID access module designed to process the streams of tag data coming from one or more readers. RFID access module performs filtering aggregation and counting of tag data, reducing the volume of data prior to sending to applications such as enterprise resource planning and supply chain management system. Security module has to provide the security mechanisms such as denial of Service tolerance, malicious request or query filtering. But as we tried to say on the statement of problem work lacks Trust worthiness of the tag which means the tag is not secure. What if the tag is prepared by attackers or cloned and used for malicious purpose there is no way defined for managing this problem.

Dimitriou [32] proposed an RFID authentication protocol (referred to here as the D protocol) in 2005, designed to enforce user privacy and protect against tag cloning. A tag T_i stores its identifier ID_i, and a server S stores the identifier Id_i and a hash of the identifier HID_i for each tag Ti, where HIDi serves as the primary key is used to identify information related to the tag. This scheme makes use of a challenge response approach and employs a hash function h and a keyed hash function f. A server queries a tag by sending it a random number r_1, and a

tag responds with a random number r_2, a hash of its identifier M_1= h (ID_i), and a keyed hash of the random numbers M_2 = fID_i (r_2kr_1).

The scheme maintains scalability in the sense that the server can find the value HIDi corresponding to the received value of M_1, without an exhaustive search. If the server finds a matching value HID_i, it checks that the received value of M_2 equals fID_i (r_2kr_1). If the validation is successful, the server authenticates the tag and updates its identifier *Idi*to g (ID_i), where g is a one way function. The server then sends a message M_3 = fID_i (r_2kr_1) using the updated identifier to the tag. The tag authenticates the server by checking the received value of M_3. If the check is successful, the copy of the identifier Id_i held by the tag is also updated.

3.2 RFID protocols using Hash Function approach

Weis et al. [7] proposed a simple security scheme based on one-way hash functions called Hash Lock Scheme. Each tag has a portion of memory reserved for storing a temporary meta-ID, and operates in either a locked or an unlocked state. The reader hashes a key *K* for each tag, and each tag holds a meta-ID (meta-ID = hash (K)). While locked, a tag answers all queries with his meta-ID and offers no other functionality. To unlock a tag, the owner queries the back-end database with the meta-ID from the tag, looks up the appropriate key and sends the key to the tag. The tag hashes the key and compares it to the stored meta-ID as depicted in Figure 3.2.

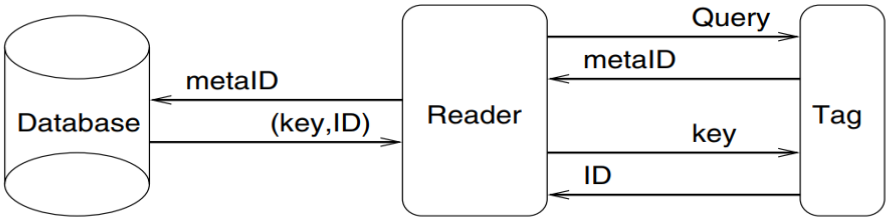


Figure 3.2: Hash-Locking, A reader unlocks a hash-locked tag

Weis also proposed an extension of the hash lock scheme [23], Randomized Hash Lock Scheme one of the problems of the Hash Lock Scheme solution is that it allows the tracking of individuals. To avoid this, the meta-ID should be changed repeatedly in an unpredictable way. In order to solve this problem, it requires that tags have a hash function and a pseudo-random number generator and method is depicted in Figure 3.3. In hash Lock based is a

scheme which involves locking a tag using a one-way hash function. A locked tag uses the hash of a random key as its meta-ID=Hash (key). When locked, a tag responds to all queries with its value of meta-ID [8, 36].

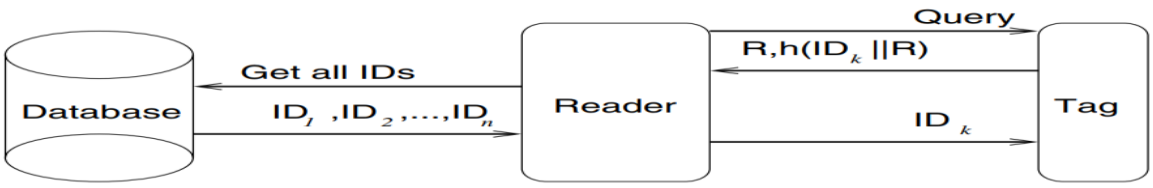


Figure 3.3: Randomized Hash-Locking, A reader unlocks a tag whose ID is k in the randomized

In [21], a hash-chain scheme was proposed in which two hash functions (G and H) are embedded in the tag. The secret key s is stored in the tag's memory, which is linked to the object's ID server, which manages the link between the secret key *s* and the object's ID. Tag responds to reader queries by generating a hash value a=G(s) of secret s, computing new secret s'=H(s), and overwriting the memory with new secret s'. The reader sends the output from the tag to the server and makes a request for revealing.

Summary

Different RFID security based works that are related and relevant to our proposed system are investigated and analyzed**.** From the analysis of the related works, we came up with the following major problems.

- In case of Synchronized Secrets, if a cloned tag enters the supply chain before the corresponding genuine tag is read the cloned tag will first go unnoticed and the alarm will be triggered when the genuine tag is read the next time. As a result, the counterfeit product can already be consumed before the alarm is triggered. In addition, the synchronized secrets method needs to know when the tagged products leave the RFID system to "close the trace". As a result, the method is vulnerable to injection of unnoticed cloned tags if it is not known when the genuine products are no longer within the traced system.
- In case of D protocol however, the tag identifier remains the same between valid sessions, thereby making the scheme vulnerable to tracking. Additionally, the

scheme is prone to DoS attacks [13], if the message M_3 does not reach the tag in a session, the server will update the tag identifier but the tag will not.

- Previous scheme allows a tag to be tracked because the same meta-ID is used repeatedly.
- There are different hash functions based solution proposed to perform encrypt operations like SHA-1, SHA-256, AES, MD4 , MD5 etc. problems with all this algorithms is the computation overhead and power requirement for total number of operations. And, Most of the cryptographic techniques mentioned are applicable for only active tags as they requires more computing as illustrated in Table 3.1.

Table 3.1: RFID Tag Gate requirement and cryptographic algorithm

Algorithm	Chip Areal Equivalent
SHA-256	10,868 Gates
SHA-1	8120 Gates
MD5	8400 Gates
MD4	7350 Gates
AES	3400 Gates

CHAPTER 4: THE PROPOSED SOLUTION

4.1 Introduction

The content in the RFID tag never changes until next encoding; the ability to easily read it will enable several security risks. First, the adversary may determine which object owns the tag and infer the origin of the object carrying the tag. Second, any static identifier can be used both to track and identify tagged objects. The main security risk of the static tag content is the leakage information and possibility of cloning the tag.

Because of the static properties, RFID tag content always remains unchanged after scanning is completed, and the attacker conducts multiple actions on the target tag. As we stated above tag that store static tag content is vulnerable to attacks such as cloning the tag so our proposed method comes with the idea of storing dynamic value to the tag which is making the tag content dynamic by changing the content and synchronizing it with the information known by the system every time the tag is read/scanned. The back-end system keeps track of which number is written on which tag.

When a tag is scanned its Tag_id is updated both on the tag and the back-end. As illustrated in Figure 4.1, layered system architecture is proposed for managing RFID data. The structure of the proposed system is broken down into three subsystems. Tag Management module, Read/Write module and Database module. The physical environment consists of RFID tags attached to different objects. The next layer, called Read/Write layer, is the layer of RFID reader which captures data from the tag. The data emerging from this layer can be considered as RFID data streams. The third layer of the system architecture is Tag management module the layer which helps us to interface reader and database. It is a middle module in the systems typically deployed between the reader and database in order to take captured data from reader and send this data to the database then send back to the reader new tag_id to reader after authentication process is done. Authentication process is done by matching received tag_id with the stored tag_id. It provides connectivity with RFID reader (via the reader adapter), processing raw RFID data for consumption by applications and providing an application level interface to manage tag and captured data. In the next subsection, we describe each module of the system in detail.

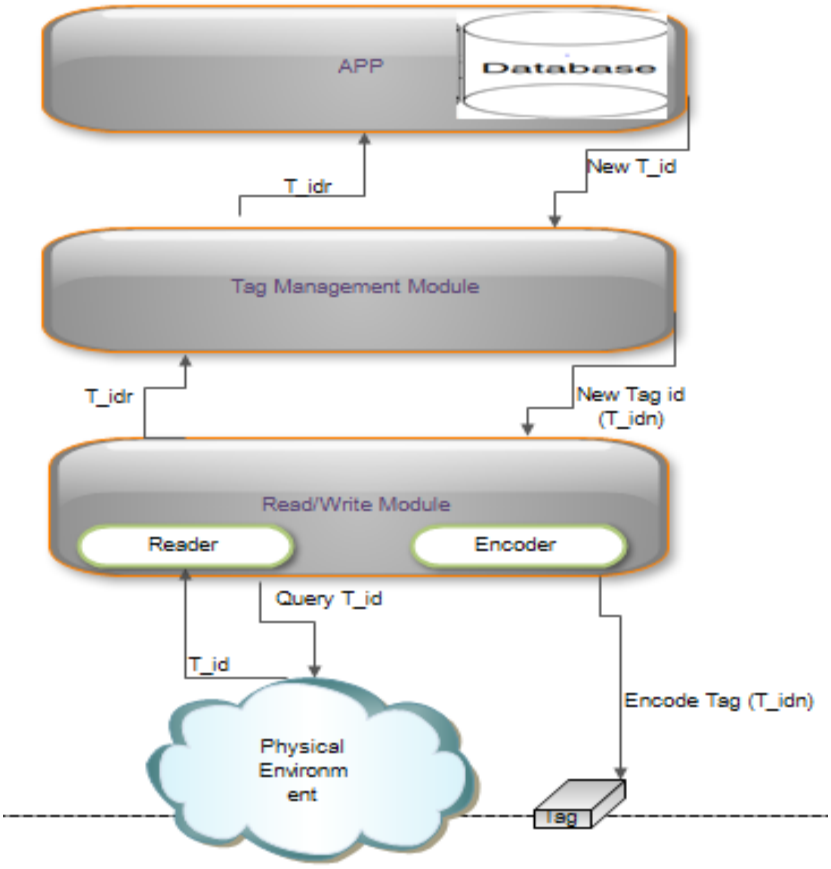


Figure 4.1: System Architecture

4.2 Architecture components

a. Tag Management Module

RFID Tag Management module is the interface that operates between the RFID hardware and RFID applications. It focuses on processing the RFID data before it is routed to the specific applications. It collects and passes the readings from readers for the use of enterprise applications and enterprise database, algorithm 4.1 illustrates this. On each read action done by the reader, Tag Management module receives tag content from the reader and sends this data to the database. In the database the tag_id is checked for its existence. If the tag_id exists, access or any further action is granted for the tagged objects. At the same time a new tag_id is selected and sent to the Read/Write module for encoding. Algorithm 4.1 demonstrates encoding tag with new tag_id. But, if the tag_id does not exist in the database, access is not granted for the tagged object and authentication flow is given in algorithm 4.2.

```
Function EncodeTID (TIDr)
begin
    TIDn=ReceiveTID
    Transmit TIDn to the reader:
     encode TIDn to the tag;
end function
```

Algorithm 4.1: Encode tag_id

```
Function AuthenticateTag(TIDr)
   begin
      If (TIDR== TIDD)
          Authentications pass;
      Else
          Authentications fail;
end function
```

Algorithm 4.2: Authentication

b. Read/Write Module

Read/Write module initiate, the action and requesting tag. After reading the tag_id, it sends the value to the Tag Management module. It waits for the new tag_id then after authentication Tag Management module sends the newly selected tag_id to the Read/Write module. Finally the Read/Write module encodes tag_id to the tag. Figure 4.2 illustrate the Read/Write module and the flow is presented in algorithm 4.3.

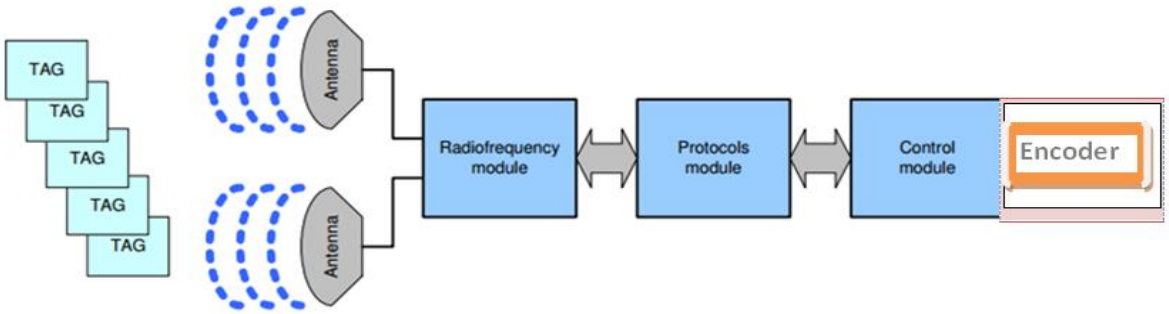


Figure 4.2: Read/Write module

```
Function redTagvalue ()

    begin

      TIDr= redTagvalue ()

       Return TIDr

end function
```

Algorithm 4.3: Read command for reader

c. Database Module

Application and database are the top most layers of components in RFID system architecture. This layer converts the data from middle-module into meaningful information. It then delivers the information to other application system so every time a tag is read; Read/Write module sends the value to the tag management module. This module sends the received value to the back end database then the back-end first verifies the tag. If this number is valid and exists in the database access is granted for the tagged object otherwise access is denied. After the check, the back-end generates a new tag_id (tag_idn) that the reader device writes on the tag. The database deletes old tag_id after access is granted for the tagged object. New tag selection is depicted in algorithm 4.4.

4.3 Rules for Generating New Tag_id

1. Should be unique in the data base.

2. Old Tag_id must be different from new tag_id (T_ido ≠ T_idn)

3. NewTag_id =Max (Tag_idofDB) +1.

```
Function SelectNewTID()
beging
    TID=ChooseTID from db where TIDr==TIDd
        If(TID==TIDdmax+1)
          TIDn=TID
        Else
        TIDn=TIDr
end function
```

Algorithm 4.4: New tag_id selection

CHAPTER 5: IMPLEMENTATION AND EXPERIMENTATION

This Chapter describes the implementation and demonstration of secure RFID system whose architecture is discussed in Chapter four.

5.1 Implementation

5.1.1 Development Tools

We decided to implement and simulate the proposed system using Arduino IDE, proteus simulations tool and Java programming tools.

Proteus

Proteus Virtual System Modeling combines mixed mode spice circuit simulation, animated components and microprocessor models to facilitate co-simulation of complete microcontroller based designs. It makes it possible to test designs before construction of prototype. This is made possible by interaction with design using on screen indicator such as leds, lcds, and actuators such as switches and buttons.

We select Proteus because

- ✓ It is open source simulation and designing software.
- ✓ Draw schematics and simulate the circuits in real time.
- ✓ The simulation allows human access during run time, thus providing real time simulation.

5.1.2 Prototype

a. Simulation and discussion

As shown in design part, our system consists of read/write module so in replacement of this module we use key pad and led for our simulation then when RFID tag placed on the RFID reader then it read and send to the controller. The controller match this received content with stored one. Old tag_id is store in EEPROM (Storage layer of proposed architecture). It receives new tag_id interred with the help of microcontroller. If the content matches with the stored one then access is granted unless it denies access as illustrated in algorithm 5.1.

```
Function GetTagvalue ()
    Begin
        Read tag ID
        TID= GetValues (TID_R);
    Server verifies TID within the DB;
            If (TID_R== TID_D)
                Authentications pass //Green LED turn on
                Else
                Authentications fail //Red LED turn on
End function
```

Algorithm 5.1: Proteus Simulation using one stored tag_id

When led turn to green implies that the tag-id is legitimate. The tag_id holder obtains the echo/pop up message saying change tag_id. He/she inserts the old tag_id before inserting new tag_id. Microcontroller checks the presence of the user by matching inserted tag_id with stored one. If it is correct it let them change their tag_id unless otherwise ask the user to insert correct tag_id. Figure: 5.1 present the Circuit Diagram that is used for simulation with each component and the output of the simulation is given in annex part.

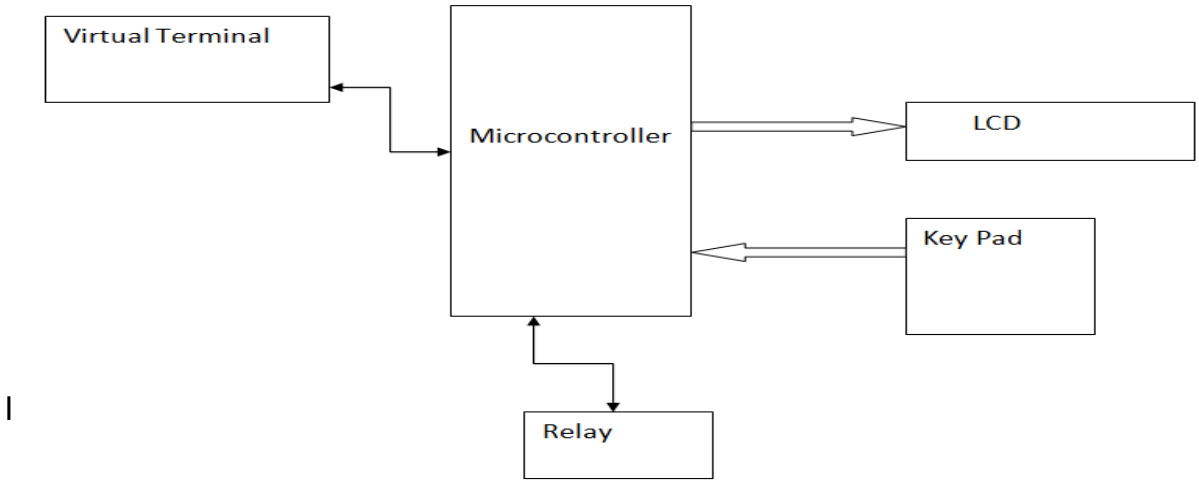


Figure 5.1: Circuit Block Diagram

b. Reading, writing and synchronizing multiple tag-id

The reader reads tag contents continuously. After reading is done it checks if there is new tag interring the system every three seconds. If new tag inters the system the reader reads the content and store it then starts synchronize. As a result integrity is preserved and the cloned tag can be identified easily.

To synchronize TIDs in the database1 with newly entered TIDs. First TIDs stored in database1 should be stored to database2. This is done by reading TIDs in database and storing one by one to database2. After reaching the end of database one the algorithm checks if there are new entries. If there are new entries it will write to database2 then synchronize. Synchronization starts after reading all the TIDS in database including the new entries. The algorithm checks for the presence of each TID in the two databases. For each TID in database1 it searches for the matching TID in database2. When it finds the match it will stop and go to the next TIDS until it reaches the end. Finally the algorithm returns synchronized if all of the tag_id matches, algorithm 5.2 illustrate the method used.

```
Open TID database1

Open TID database1

TID1 ←-------- TIDs in database1

TID2 ←-------- TIDs in database2

Result←--------String to store output

Size ←-------- Numbers of TID stored, initialized as 0

Count ←------Integer to store Number of synchronized TID's

Found←------Boolean to indicate presence of TIDs

While end of database1 not reached

    R: read each TID1 from database1

        For each TID1

            Write TID1 to database2

 Check for new entry

            While end of new entry not reached

                Go to R

For each TID1 in database

        While (found== false)

            For each TID2 in database2

            if (TID1== TID2)

            found= true
```

```
                    count++

if (count==size)

        result= synchronized

else

        result =desynchronized

  return result
```

Algorithm 5:2: Simulation for reading, writing and synchronizing data

CHAPTER 6: CONCLUSION AND RECOMMENDATION

6.1 Conclusion

We started out with a comprehensive assessment of the state of the art concerning with RFID technology, RFID systems, components and communications methods as well as the main standards related to RFID technology then study and find out the main attacks on RFID systems and also we listed and analyzed the main works done for securing RFID technology. Finally new idea and layered architecture for improving RFID security is designed.

RFID is now a topic of interest in a great number of works. Many of these works focus on security, and the range of the proposals is very wide. Some authors have proposed solutions based on cryptographic techniques. These solutions are very diverse, some of them being based on block-ciphers, pseudo-random number generators, and even public-key cryptography. However, the most commonly proposed solution is based on hash functions.

All of these protocols share the common characteristic of being single round protocols. In a different approximation, the family of human based protocols is based on multiple execution of a very simple round. The majority of proposals to make RFID tags secure make two important errors. First, they propose a protocol for RFID tags without specifying for which class of RFID tag the protocol is intended. This is a very important point, as the number of available resources (memory, circuitry, power, etc.) will highly depend on this. Thus not all tags will support the same kind of operations. Additionally, each RFID class should have a different security level. Secondly, the proposed protocols are not realistic about tag resources. As we have already mentioned, the most widely-adopted proposal is based on hash functions. In spite of this, many authors claim that their protocols are appropriate for low-cost RFID tags. However, a maximum of 4K gates can be devoted to security functions in this class of tag. As we saw in related work, considerable resources (over 9K gates) are needed to implement traditional cryptographic hash functions.

We proposed dynamic tag_id management that is applicable for any kind of RFID tag in order to address the gaps identified in the previous works and to solve RFID security. The different components of the proposed architecture and their operations, as well as interactions are presented in depth in Chapter 4. It includes: application module, tag

management module and Read/write module which consists of reader and encoder. We then implemented our work and try to demonstrate the proposed solution.

6.2 Contribution

- We designed RFID security that is free of environmental factors like noise, temperature etc.
- The proposed security mechanisms do not need extra cost for implementing for RFID tag.
- No need of extra storage space is required.

6.3 Recommendations

An important part of our research activity is centered on static tag_id implementation for RFID tags. A secure RFID application must be backed by a secure back-end server. It is the first defense of the whole system. We suggest the back-end server to be non-comprisable. Also, the communication channel between the reader and the back-end server which is database has to be secured or else all the security measures implemented at the tag level will be useless. For the above requirements, we suggest the following points.

- To guarantee the first defense, the back-end database server is better to stay private and not to be publicly accessible.
- Those who can access the back-end database server must be legitimate. A security mechanism is suggested to authorize and authenticate all the connecting readers (or operators if human is involved) such that no unauthorized access is allowed.
- There must be protection to protect curious or even malicious outsiders from accessing the legitimate readers easily. It is suggested to have another system to monitor the use of legitimate readers, by whom, when and where. This also helps guarantee the honesty of the back-end system.
- It is suggested to keep log of all the access to the back-end database server (e.g. access time, reader ID, queried information, etc.) such that any adversarial activities can be traced.
- In case there is a legitimate reader being stolen, there must be some contingent plans to stop the stolen reader from accessing the backend database server again.

6.4 Future Work

There are additional applications areas beyond dynamic tag_id management such as the protocols that are focused on the problem of providing a proof for the simultaneous reading of two or more RFID tags. Our work does not consider securing the tag during simultaneous reading we just tried to synchronize tags_id during simultaneous reading. A close inspection of this issue should be done in association to dynamic tag_id management.

REFERENCES

[1] M. Weiser, "The computer for the 21st century", *Scientific American* Vol. 265, No. 3, pp. 94–104, 1991.

[2] F. Thornton, B. Haines, A. M. Das, H. Bhargava, and A. Campbell, "RFID Security" *Syngress Publishing,Inc., Rockland*, USA, 2006.

[3] L. Stegeman, "Who's Afraid of the Big Bad Wolf?" *Market Wire*, 2004.

[4] I. Vajda and L. Buttyan, "Lightweight Authentication Protocols for Low-Cost RFID Tags", *Proceedings of the 2nd Workshop on Security in Ubiquitous Computing*, pp.1-10, 2003.

[5] I. Kim, B. Lee, and H. Kim, "Privacy Protection Based on User-defined Preferences in RFID System", *International Conference on Advanced Communication Technology, ICACT'06*, 2006, pp. 858-862.

[6] H. Lee and J. Kim, "Privacy Threats and Issues in Mobile RFID", P*roceedings of the First International Conference on Availability, Reliability, and Security, IEEE Computer Society*, April, 2006.

[7] S. L. Garfmkel, A. Juels, and R. Pappu, "RFID Privacy: An Overview of Problems and Proposed Solutions", IEEE Security and Privacy, Vol. 3, 2005, pp. 34-43.

[8] G. Avoine, "Adversary model for radio frequency identification*", Technical Report lasec report, Swiss Federal Institute of Technology (EPFL),* Switzerland, September 2005. A. Juels and S. Weis, "Defining strong privacy for RFID", *Cryptology ePrint Archive Report*, 2006.

[9] T.Le, M.Burmester, and B.Medeiros," Forward secure RFID authentication and key exchange", *Cryptology ePrint Archive Report*, 2007.

[10] H. Gilbert, M. Robshaw, and H. Sibert, "An active attack against HBa provably secure lightweight authentication protocol", *Manuscrip*t, July 2005.

[11] Simson, "PGP: Pretty Good Privacy", *Reilly& Associates, Inc.*,1995.

[12] S. Weis, S. Sarma, R. Rivest, and D. Engels, "Security and privacy aspects of low-cost radio frequency identification systems", *International Conference on Security in Pervasive Computing SPC 2003,* Vol. 2802, Germany, March 2003.

[13] S. Bruce, J.Wiley and Sons, "Applied Cryptography", 1996.

[14] M. Mostafa, E. Said, and I. Wood ring, "An Empirical Study for Protecting Passive RFID Systems against Cloning," in *Sixth International Conference on Information Technology*, Las Vegas, 2009.

[15] M. Mikko and Lehtonen, "Securing RFID systems by detecting tag cloning," *Pervasive Computing*, pp. 291-308, 11-14 May 2009.

[16] G. Michael, "Wal-Mart reading RFID tags in Texas", *Supermarket News*, Vol. 52, No. 19, p. 61, 2004.

[17] R. M. Chris, "Radio Frequency Identification (RFID)," *Computers & Security*, vol. 25, No. 1, pp. 18-26, 2006.

[18] R. George, "Networked RFID systems, software and services*," London, Springer*, 2008.

[19] Duc D. N., Park J., Lee H., "Enhancing security of EPCglobal GEN-2 RFID tag against traceability and cloning", *in Proceedings of the Symposium on Cryptography and Information Security (SCIS 2006) Hiroshima Japan*, pp.17-20, 2006.

[20] Sarma S. E., Weis S. A., Engels D. W., RFID systems and security and privacy implications, *Lectures Notes in Computer Science* 2523, 2003, pp. 454-469.

[21] A. Ilic, M. Lehtonen, F. Michahelles and E. Fleisch, "Synchronized Secrets Approach for RFID technology|", *Information Management Conference*, January 20011.

[22] M. Ohkubo, K. Suzuki, and S. Kinoshita, "Cryptographic approach to privacy-friendly", *in Proc. of RFID Privacy Workshop*, 2003.

[23] D. M. Konidala and K. Kim, "RFID tag-reader mutual authentication scheme utilizing tag's access password" *Auto-ID Labs White Paper WP-HARDWARE-033*, 2007.

[24] J. Lou, G. Andrechak and M. Riben,"" A Review of Radio Frequency Identification Technology for the anatomic pathology, Aug 13, 2011.

[25] J.Res, "Radio Frequency Identification (RFID) technology and patient safety", v.18 (9), Sep 2013.

[26] Jung, T. Koo, and J. Kim. "RFID system recognition test under the forestry environment for tree management", *Fifth International Joint Conference on INC, IMS, and IDC*, 2009.

[27] Korten, S and Kaul, C, "Application of RFID (Radio Frequency Identification) in the Timber Supply Chain", *Croatian J. Forest Engineering*, V.29 pp. 85-94, 2008.

[28] L. Th. Mirowski and BComp, "Detecting Clone Radio Frequency Identification Tags", University of Tasmania ,November, 2006

[29] Mostafa and Ira**,** "Introduction to RFID Technology"**,** *The Art of Identification*, 2008.

[30] D. N. Duc, J. Park, H. Lee, and K. Kim, "Enhancing security of EPCglobal gen- 2 RFID tag against traceability and cloning", In *Symposium on Cryptography and Information Security*, Japan, January 2006.

[31] T. Dimitriou, "A lightweight RFID protocol to protect against traceability and cloning attacks", In *Conference on Security and Privacy for Emerging Areas in Communication Networks*, pp.59, Greece, September 2005.

[32] R. M. Chris, "Radio frequency identification (RFID)," *Computers & Security*, Vol. 25, No. 1, pp. 18-26, 2006(as 18)

[33] Sarma S. E., Weis S. A., Engels D. W., "RFID systems and security and privacy implications", *Lectures Notes in Computer Science*, pp. 454-469, 2003(as 21).

[34] B. Julien, H. Chabanne, and I. Thomas, "Cryptanalysis r RFID identification protocol", *In Matthew Franklin, LucasHui, and Duncan Wong, editors, Cryptology and Network Security*, v. 5339, pp. 149, Springer Berlin ,2008.

[35] A. Weis, E. Sarma, L. Rivest, and W. Daniel , "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems", *Auto-ID Center Massachusetts Institute of Technology Cambridge.*

ANNEXES

Annex A: Java Code that read writes and synchronize contents continuously

```
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.io.RandomAccessFile;
public class ReadAndAppendContinually {
	static int syncNextLineNumberToRead = 1;
	static int loop1NextLineNumberToRead = 1;
	public static void main(String[] args) {
		loop1();
	}
	static void sync(){
		RandomAccessFile br = null;
		String fileName = "C:\Users\bemenet\Desktop\rfid
data\ReadWriteAppend\src/data.txt";
		try {
			br = new RandomAccessFile(fileName, "r");
		} catch (FileNotFoundException e1) {
			System.out.println(fileName + " not found");
		}
		String line;
		for (int i = 1; i < syncNextLineNumberToRead; i++) {
			try {
		   String readLine = br.readLine();
			} catch (IOException e) {
				e.printStackTrace();
			}
		}
```

```
		while(true){
			try {
				line=br.readLine();
				if(null != line){
					System.out.println("TAG_ID " +
syncNextLineNumberToRead + "- "+line );
					loop2(Integer.valueOf(line));
					System.out.println("Sleeping 2 seconds" );
					Thread.sleep(2000);
				}else{
					br.close();
					System.out.println("Sleeping 3 seconds" );
					Thread.sleep(3000);
					sync();
				}
				syncNextLineNumberToRead++;
			} catch (IOException | InterruptedException e) {
				e.toString();
				e.printStackTrace();
			}
		}
	}
	static void loop1(){
		RandomAccessFile br = null;
		String dataFileName = "C:\Users\bemenet\Desktop\rfid
data\ReadWriteAppend\src/data.txt";
		try {
			br = new RandomAccessFile(dataFileName, "r");
		} catch (FileNotFoundException e1) {
			System.out.println(dataFileName + " not found");
			e1.printStackTrace();
```

```
                }
                String line;
                for (int i = 1; i < loop1NextLineNumberToRead; i++) {
                        try {
                                br.readLine();
                        } catch (IOException e) {
                                e.printStackTrace();
                        }
                }
                while(true){
                        try {
                                line=br.readLine();
                                if(null != line){
                                        System.out.println("TAG_ID " +
loop1NextLineNumberToRead + "- "+line );
                                        //loop2();
        System.out.println("Sleeping 2 seconds" );
                                        Thread.sleep(2000);
                                }else{
                                        br.close();
                                        System.out.println("Sleeping 3 seconds" );
                                        //Wait 3 seconds before checking the file again
                                        Thread.sleep(3000);
                                        //Recursively call my method to check if there is a new
tag content added after the sleep
                                                sync();
                                }
                                loop1NextLineNumberToRead++;
                        } catch (IOException | InterruptedException e) {
                                e.printStackTrace();
                        }
```

```
		}
	}
	static void loop2(Integer inputIntegerToAppend){
		BufferedWriter bWriter = null;
		FileWriter fWriter = null;
		String appendedFilename = "C:\Users\bemenet\Desktop\rfid
data\ReadWriteAppend\src/appendedData.txt";
		try {
			File fileToWriterTo = new File(appendedFilename);
			//if file is not found, we create a new one, good for first time
			if (!fileToWriterTo.exists()) {
				fileToWriterTo.createNewFile();
			}
			fWriter = new FileWriter(fileToWriterTo.getAbsoluteFile(),true);
			bWriter = new BufferedWriter(fWriter);
			bWriter.write(String.valueOf(inputIntegerToAppend));
			bWriter.newLine();
			System.out.println("Completed writing tag content:
"+inputIntegerToAppend );
		} catch (IOException ioe) {
			ioe.printStackTrace();
		}finally {
			try {
				if(bWriter != null){
					bWriter.close();
				}
				if(fWriter != null){
					fWriter.close();
				}
```

```
                    } catch (IOException ioe) {
                        ioe.printStackTrace();
                    }

                }

            }


}
```

Annex B: Arduino Code used for simulation

```
#include <Keypad.h>
#include<LiquidCrystal.h>
#include<EEPROM.h>
LiquidCrystal lcd(9,8,7,6,5,4);
char password[3];
char pass[3],pass1[3];
int i=0;
char customKey=0;
const byte ROWS = 4;
const byte COLS = 4;
char hexaKeys[ROWS][COLS] = {
{'1','2','3','A'},
{'4','5','6','B'},
{'7','8','9','C'},
{'*','0','#','D'}
};
byte rowPins[ROWS] = {A0,A1,A2,A3};
byte colPins[COLS] = {A4,A5,3,2};
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS,
COLS);
int led = 12;
int leds = 13;
int buzzer = 10;
int m11;
int m12;
void setup()
{
  Serial.begin(9600);
  pinMode(11, OUTPUT);
  lcd.begin(16,2);
```

```
  pinMode(led, OUTPUT);
  pinMode(leds, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(m11, OUTPUT);
  pinMode(m12, OUTPUT);
  lcd.print("RFID");
  lcd.setCursor(0,1);
  lcd.print("Basedaccess");
  Serial.print("RFID security");
  delay(500);
  lcd.clear();
  lcd.print("Enter tag_id");
  lcd.setCursor(0,1);
  for(int j=0;j<3;j++)
    EEPROM.write(j, j+49);
  for(int j=0;j<3;j++)
    pass[j]=EEPROM.read(j);
}
void loop()
{
  digitalWrite(11, LOW);
  customKey = customKeypad.getKey();
  if(customKey=='#')
    change();
  if (customKey)
  {
    password[i++]=customKey;
    lcd.print(customKey);
  beep();
  }
  if(i==3)
```

```
{
 delay(500);
 for(int j=0;j<3;j++)
   pass[j]=EEPROM.read(j);
 if(!(strncmp(password, pass,3)))
 {
   digitalWrite(led, HIGH);
beep();
   lcd.clear();
   lcd.print("Get in");
   digitalWrite(11, HIGH);
   delay(500);
   lcd.setCursor(0,1);
   lcd.print("#Change tad_id");
   delay(500);
   lcd.clear();
   lcd.print("Enter tad_id");
   lcd.setCursor(0,1);
   i=0;
   digitalWrite(led, LOW);
   digitalWrite(leds, LOW);
 }
 else
 {
   digitalWrite(11, LOW);
   digitalWrite(buzzer, HIGH);
   digitalWrite(led, LOW);
   digitalWrite(leds, HIGH);
   lcd.clear();
   lcd.print("WRONG tad_id");
   lcd.setCursor(0,1);
```

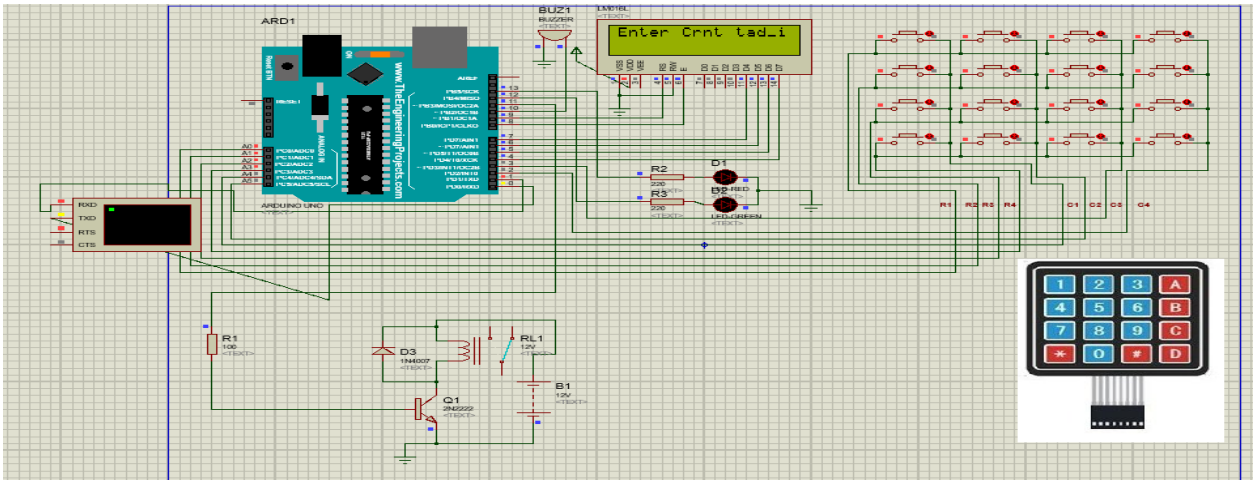
```
      lcd.print("#Change tad_id");
      delay(2000);
      lcd.clear();
      lcd.print("Enter ur tad_id:");
      lcd.setCursor(0,1);
      i=0;
      digitalWrite(buzzer, LOW);
      digitalWrite(led, LOW);
      digitalWrite(leds, LOW);
    }
  }
}
void change()
{
  int j=0;
  lcd.clear();
  lcd.print("Enter Crnt tad_id");
  lcd.setCursor(0,1);
  while(j<3)
  {
    char key=customKeypad.getKey();
    if(key)
    {
      pass1[j++]=key;
      lcd.print(key);    }
    key=0;
  }
  delay(500);

  if((strncmp(pass1, pass, 3)))
  {
```

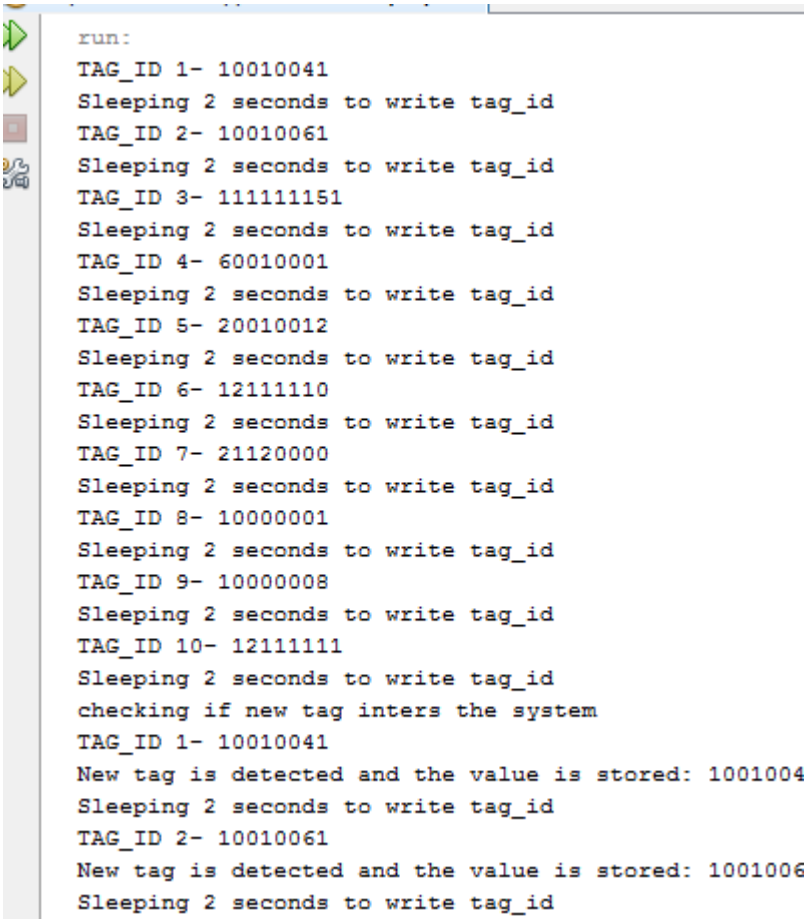
```
  lcd.clear();
  lcd.print("WRONG tag id...");
  lcd.setCursor(0,1);
  lcd.print("Try Again");
  delay(500);
 }
 else
 {
  j=0;
  lcd.clear();
  lcd.print("Enter new tag id:");
  lcd.setCursor(0,1);
  while(j<3)
  {
   char key=customKeypad.getKey();
   if(key)
   {
    pass[j]=key;
    lcd.print(key);
    EEPROM.write(j,key);
    j++;
   }
  }
  lcd.print(" Success..");
  delay(1000);
 }
 lcd.clear();
 lcd.print("Enter tag id:");
 lcd.setCursor(0,1);
 customKey=0;
}
```

```
void beep()
{
  digitalWrite(buzzer, HIGH);
  delay(20);
  digitalWrite(buzzer, LOW);
  }
```

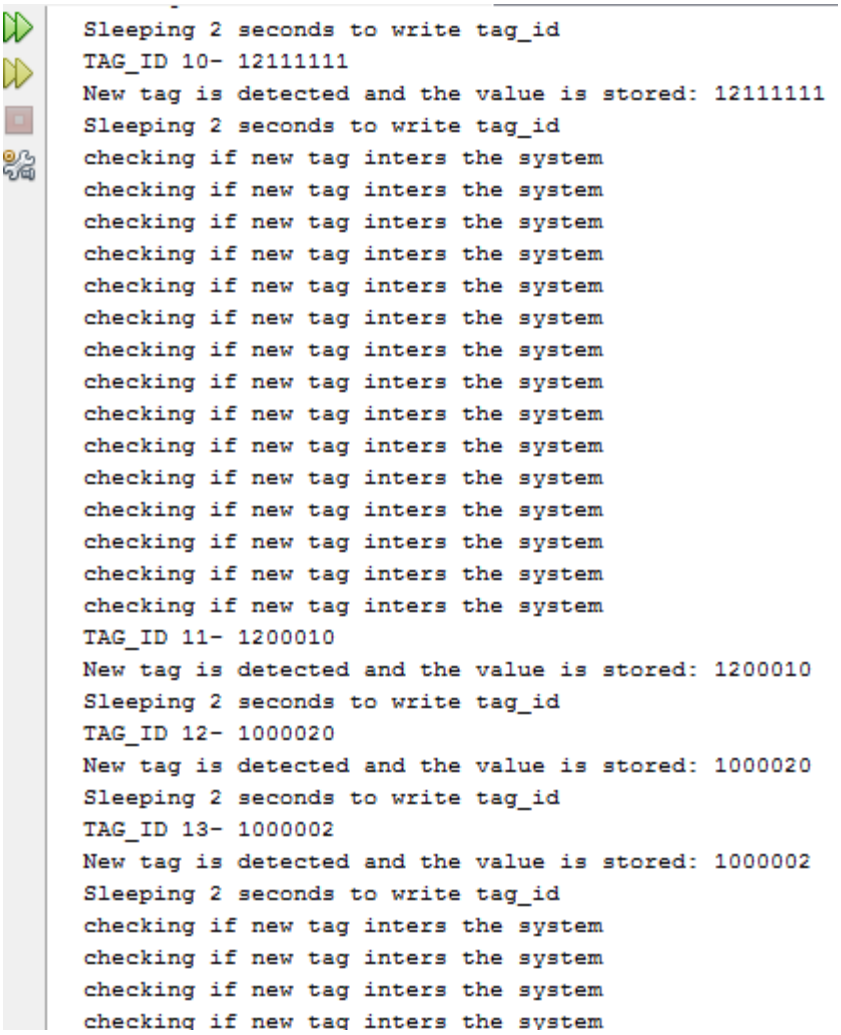
Annex C: Simulation for asking user for current tag_id for authentication before changing new tag.



Annex D: Simulation for reading, writing and synchronizing data



```
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
TAG_ID 16- 1000221
New tag is detected and the value is stored: 1000221
Sleeping 2 seconds to write tag_id
TAG_ID 17- 1222222
New tag is detected and the value is stored: 1222222
Sleeping 2 seconds to write tag_id
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
TAG_ID 18- 1122334
New tag is detected and the value is stored: 1122334
Sleeping 2 seconds to write tag_id
TAG_ID 19- 1111111
New tag is detected and the value is stored: 1111111
Sleeping 2 seconds to write tag_id
TAG_ID 20- 1122333
New tag is detected and the value is stored: 1122333
Sleeping 2 seconds to write tag_id
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
```


Sleeping 2 seconds to write tag_id
TAG_ID 10- 12111111
New tag is detected and the value is stored: 12111111
Sleeping 2 seconds to write tag_id
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
TAG_ID 11- 1200010
New tag is detected and the value is stored: 1200010
Sleeping 2 seconds to write tag_id
TAG_ID 12- 1000020
New tag is detected and the value is stored: 1000020
Sleeping 2 seconds to write tag_id
TAG_ID 13- 1000002
New tag is detected and the value is stored: 1000002
Sleeping 2 seconds to write tag_id
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system
checking if new tag inters the system

Declaration

I the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all source of material used for the thesis have been duly acknowledge

Declared by:

Name: Bemenet Kassahun

Signature: _____

Date: _____

Confirmed by advisor:

Name: Dr, Dagmawi Lemma

Signature: _____

Date: _____