Abstract

Radio Frequency Identification (RFID) is considered an emerging technology that offers significant benefits to organizations, industries, and even individuals. A typical RFID system consists of a reader, tags and the back-end server that accomplish the automated identification of objects, humans, livestock etc requiring no physical contact. Accompanying with the technology's praises follow the alarming echoes affiliated with the level of privacy and security the technology entails. As a result, numerous recent and relevant literatures focus on the topic of "RFID security and privacy". However, no long-term security solution has yet been proposed and remains a current issue with severe impacts on RFID technology, especially when considering that the wide and fast developments of such systems are already proceeding.

This project firstly presents an overview of the technology's fundamental following unto the identification of the privacy and security issues. Several previously proposed protocols are analysed and compared, assessing their achieved level of privacy and security. Finally, with the aid of the evaluated prior art, a new protocol is presented with the corresponding evaluation.

The final section of this project presents an alternative method that ensures the secure communication between tags and readers via a privacy-preserving protocol, based on public-key cryptography. The underlined outcome of this project is interesting since as seeing throughout the literature review that focus on securing RFID tags, the majority emphasises on techniques such as symmetric-key, hash functions, XOR operations, random number generators and other similar methods. Scalability, which is another major issue, cannot be achieved if a protocol applies linear search in order to identify a tag and many previously proposed protocols either disregard this property or apply methods that require constant time, resulting in impractical solutions. However, if one investigates both the past and present state of the technology, apparent evidence shows that security threats remain within an RFID system that induce privacy issues as well. Research that emphasises on solving the current issues in the RFID technology can be characterized as an "invariable circle". This circle begins with the proposal of a secure protocol, based upon the improvements of previous protocols, and eventually the present protocol will be the subject of a forthcoming newer protocol. Researches rarely considered public-key cryptography (PKC) as a solution for RFID tags, since it is claimed to be expensive. Consequently, trying to avoid this direction, might lead authors to present protocols, which are as expensive as PKC schemes and in the mean time insecure. Nevertheless, PKC might finally be a convenient solution to ensure the highest level of privacy and security for RFID tags.

The main contributions and achievements of the project are:

- This project consists of a background research of twelve previously proposed protocols, evaluating and comparing them in terms of their achieved security and privacy properties.
- I analysed the privacy model by Vadenay [56], see pages 56, and adapted it for the proposal of the new protocol.
- I analysed Feldhofer et al. [45] version of AES encryption for low-cost tags and Lee et al. [44] elliptic-curve based security processor for passive tags. This enhanced the performance evaluation of the proposed protocol, which is based on ECIES. See pages 56 - 62.
- I presented an RFID protocol that resists tracking, replay attacks, denial of service attacks, cloning attacks and preserves forward privacy, tag anonymity and user data confidentiality. See page 63.

Acknowledgements

I would like to express my gratitude to,

Dr Steve Gregory, a kind, patient and reliable supervisor that has supported me throughout the entire MSc.

Prof. Nigel Smart, an influential and brilliant man that directed me with his knowledgeable comments during the dissertation.

Mr Andrew Charlesworth, for providing me with a sufficient amount of reading material.

My closest friends, which even though faced their own difficulties, are always there and brighten my life.

To my parents, their love and care has sustained me throughout my life. Thank you!

And finally, "O Lord, you are my God; I will exalt you and praise your name, for in perfect faithfulness you have done marvellous things, things planned long ago."

Table of Contents

1. Introduction

Radio-Frequency Identification, (RFID) is an emerging technology for the automated identification of objects, people, or animals attached to RFID tags. The technology consists of radio waves that communicate between a transponder that contains a microchip (i.e. tag) and a transceiver (i.e. reader) that identifies tags without involving physical contact and/or line-of-sight. Also characterized as the "Silent revolution", RFID is reducing costs, optimizing business procedures and will soon replace the Universal Product Code (UPC) which came about in the early 1970s [6]. Line-of-sight, limited number of item scans at a time, and other limitations exists in the UPC technology. RFID has the ability to store and retrieve multiple data automatically from tags, including unique ID numbers and other information relative to the items, livestock, or even humans that are enabled with tags. With the knowledge of the identity, location, and condition of products, as well as, animals, assets, people, identity documents and other elements; individuals, governmental units, and companies can benefit in numerous ways i.e. reduce cost, time, complexity of their processes, increase efficiency and visibility.

Many executives and researchers describe Radio Frequency Identification (RFID) as an "emerging, cutting-edge technology". Following on from the history of RFID up to the present, it is a fact that various sectors and services have already deployed some type of an RFID system or eventually will, unable to ignore the positive impacts the technology has to offer. Some of these areas include, libraries, tracking of products and livestock, supply chain management, transport payments, access control, automated vehicle identification, medical applications and patient identification, airports and e-passports.

However, apart from the significant benefits RFID technology offers, several security, and privacy issues compel solutions. Tags are considered the 'weakest link' in an RFID system due to their limited amount of resources i.e. low computational power and limited memory capacity. Owing to the mass distribution of RFID tags, their design aims to keep within low-cost boundaries, typically around 5 cents [45]. For these reasons, existing security technologies cannot be applied [73, 83, 77, 76, 82] and with the exploitation of low-cost RFID tags, arise new areas for research. A variety of possible security threats include, eavesdropping (skimming), unauthorized access and modification of data, cloning, snooping, denial of services (DoS) and the most discussed, tracking [15, 16, 17, 18, 19, 21].

The above threats are mostly formed in the air interface i.e. wireless radio communication channel between an RFID reader and an RFID tag. The insecure channel is susceptible to eavesdropping, which can lead to the manipulation, deletion, or alternation of data that affects the smooth running of the system. Another issue that provokes security threats is the unique value i.e. ID which is stored on every tag. However, other secret information stored on tags, e.g. secret keys, passwords, pins etc are also vulnerable to disclosure. For example, tags emit their unique ID to identify with the reader and if the value is fixed, then an adversary can track the tag and breach the user's privacy.

RFID privacy and security are considered a stimulating topic for researchers [17, 18]. Researchers are constantly practising cryptographic approaches, acknowledging tag capabilities and real-world security threats applied in weak security models, with the aim of developing an overall robust solution to these factors [11]. Various papers have been published that claim to provide solutions to the security and privacy challenges, examples [65 - 77]. However, these designs were later on proven vulnerable against a number of attacks rating them as overall weak designs, and neglecting one or more security properties or exciding performance boundaries [77 - 83]. The core aim of these authentication/identification protocols is for tags to authenticate with the reader through the insecure communication channel and in the mean time, preventing the leakage of any secret information to an untrusted third party. One way is to apply cryptographic techniques to protect messages. For example, symmetric cryptographic schemes, hash functions, symmetric encryption algorithms, exclusive or (XOR) are commonly used [65 - 77].

Nevertheless, the enduring issues of privacy and security remain obstacles influencing the entire approval and deployment of the excessive benefits RFID technology has to offer. Hence, the project aims to propose an RFID identification protocol that meets the identified requirements. In order to achieve this aim, the project is separated mainly into two parts. The first part studies the design of previously proposed RFID security protocols, including the reasons of choosing the specific methodology and techniques used. Then these protocols are compared against each other in order to extract the strengths and weaknesses and reveal the achieved level of security and privacy. The first part will be the cornerstone for the second part, providing appropriate knowledge to facilitate the design and the development of an RFID protocol as an improvement of the existing protocols. The final part will also include a theoretical analysis of the security properties, methods and improvements of the new protocol including a discussion on the effectiveness the solution might have in a real-life RFID system.

1.1 Main Objectives

As mentioned above, this project focuses on RFID security and privacy issues and examines several protocols that claim to be secure. The main contributions of the project are as follows:

1. Understand the fundamentals of the technology.
2. Investigate RFID security protocols and identify the relevant protocols for systems that use basic RFID tags.
3. Analyse the design of each protocol and their security properties, including their weaknesses and advantages.
4. Compare the protocols against each other in order to derive significant security properties and identify limitations that will enhance the new design.
5. With the assessment of the prior art in steps 3 and 4, a new protocol design will be proposed.
6. The efficiency and quality of the new prototype design will be compared with the analysis of the security guidelines and limitations of the findings in the previous chapters.

1.2 Organisation

The remainder of the thesis is organised as follows:

Chapter 2: introduces the fundamentals of the RFID technology, providing an overview of RFID systems, the components, the history of the technology and current applications. This chapter also introduces the privacy issues and security attacks. The final section in this chapter provides a brief description of the cryptographic primitives that are used in RFID systems.

Chapter 3: this section describes twelve previously proposed RFID protocols. There descriptions include diagrams and explanations of the authentication/identification process and techniques used.

Chapter 4: assess and compares the protocols from chapter 3 against the identified privacy, security and performance requirements.

Chapter 5: introduces three fundamental topics that enhance the design of the new protocol for the following chapter. This section consists of a privacy model description that forms the foundation of the protocol, the description of an EC security processor and the implementation of the AES encryption for low-cost tags.

Chapter 6: proposes an RFID identification protocol that achieves the identified privacy and security requirements and attains scalability properties.

Chapter 7: summarises the contributions of the project, and provides directions for future research.

2. *Fundamentals*

The technology uses radio waves to identify and connect the tagged objects to the Internet, for tracking and managing the data. The three main components all RFID systems require are the transponders (tag), the transceivers (readers) and the data processing subsystem [4].

The tag is the identification device attached to the item that involves tracking and carries the identity of the object and the data. The reader is the device that recognizes the presence of an RFID tag and reads the information stored on them or writes data to the transponder, according to the functionalities of tag. Data processing subsystems exploit the data from the transceiver corresponding to the requirements of the system. For example, RFID middleware is the software used between the applications and readers that inform other systems about the presence of the tags.

2.1 RFID System

Figure 1 below illustrates the concept of s simple RFID system (Original figure [27]):

1. The transponder: generally consists of a microchip with an antenna on the item.
2. The reader: a device with one or more antennas to read data off the microchip using radio waves.
3. RFID middleware system: the information passed by the reader to a computer connected to the Internet and from there the data is managed accordingly.

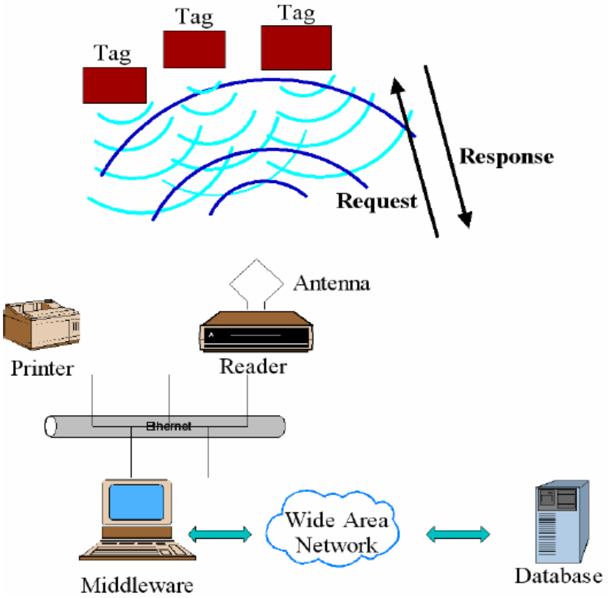


Figure 1: RFID system components

2.1.1 Tags

Tags have two general categories, active and passive. Other categories also exist and are represented in table 1. Tag types are also categorized in the form that they obtain their power source, memory capacities and communication range [1, 4]. Although, active tags and passive tags are fundamentally distinct technologies, some systems incorporate them together. Passive tags are used in applications where asset tracking is controlled and highly reliable and requires no security or memory. However, active tags are more sophisticated; for example, they best apply when businesses require complex processing of their products, security properties, on-board sensors and data management [1, 4].

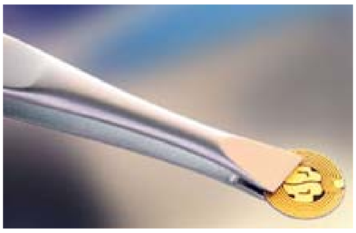
Likewise, specific tags are used in read-only and read/write systems. They defer in the sense that read-only systems are simple, requiring low-cost and power consumption. They contain an individual serial number that is transmitted to the reader when requested. These systems can be used as a replacement to the barcode systems. Read/write systems use tags that are more advanced, they contain memory, data that can be changed or updated remotely, maintenance history, content details, and other functionalities. High-end tags consist of microprocessors that include encryption and security algorithms. These tags are relevant for application that require items to be re-usable i.e. containers [1].

The table below demonstrates the EPCtag classes for RFID tags [1, 3, 6].

EPCtag[1]	Class	Functions	Uses
Class 0	Passive	Read Only	1. Can replace barcodes
Class 1		Read, write-once	1. U.S driver's licence 2. Key card 3. Parts and inventory
Class 2		Read/write Memory User data Encryption	1. E-Passport 2. Credit Cards 3. National Ids
Class 3	Semi-passive	Same as Class 2 Extras: 1. On-board power source 2. Increased range 3. Integrated sensors 4. Rewritable	1. Containers, storage, large business merchandise tracking.
Class 4	Active	Same as Class 3 Extras: 1. Self-powered active communication 2. Inter-tag communication ("two-way" tags that talk to other tags) 3. Rewritable	1. Car key fob 2. Animal tracking
Class 5		1. Powers and reads class 1, 2, 3 2. Reads class 4 and 5 3. Same functionalities as class 4	1. Highway automated toll collection

Table 1: Tag classes

Depending on the application that uses an RFID system, tags came in various shapes and sizes. Below are three examples of RFID tags (Original images [61, 62, 63]):



2.1.2 Readers

The components of an RFID reader includes the antennas for sending and receiving signals, a transceiver, wireless connectivity and a processor to decode the information captured by the tags.

[1] Non profitable organization developing commercial, world-wide RFID standards

Readers are either self-powered or powered by the device they are mounted on and are connected to a host computer that receives the captured data from the tags to the readers, for further management. According to the requirement specifications and location, either readers can be portable handheld devices or fixed devices positioned at specific points e.g. gate readers, store entrances [1, 4].

RFID readers use radio frequency (RF) transmission energy to obtain information from the tags[2]. Both tags and readers have antennas that allow them to communicate with each other through RF communication. In more detail, when RFID tags are within the range of the reader in passive systems, tags use their antenna to absorb the transmitted energy from the reader and once the tag is activated, it sends back its ID number and other necessary information. In an active system, tags transmit signals occasionally and multiple readers then capture the data [1, 4].

RFID readers come in two classes; read-only (for passive tags) and read/write (for semi-active, active tags) which update/add or remove data from the tags (requires memory). The difference between a reader and an interrogator is the type of system in which they are utilized. Interrogators are used in read/write systems whereas a reader is a read-only device [1, 4].

Example of RFID Readers (Original images [59, 60])

2.1.3 RFID Middleware

RFID middleware is the software tool that merges RFID hardware with IT systems and routes the data to companies and organisations. It is responsible for the management and quality control of the information obtained from RFID systems. The components of the middleware can be positioned at different locations than the actual system. I.e. if the RFID system is set up inside a store, the middleware might be positioned in the factory [6].

According to Hunt et al. [6], the functions of RFID middleware are separated into four categories. Data routing: middleware routes the data to the appropriate company or organisation that the system belongs to i.e. the network of the reader collects data and the middleware directs it to a warehouse that manages the tracking of the tagged items. Process management: middleware manages circumstances during business processing events, such as sending the appropriate information to be handled by the tags. Data collection: multiple RFID readers collect raw information and then the middleware software is responsible to extract and filter the information so that the IT systems obtain the right format of the data and from there manage it accordingly. Device management: large enterprise companies employ a wide diversity of readers and tags types, working on different functionalities and frequencies. Middleware coordinates and observes the status of such systems, to provide a harmonious flow of cooperation between the components.

[2] Note: readers usually work on one specific radio frequency and applications that employ tags from different manufacture might use different frequencies, thus forcing the user of the RFID system to cover their facility with multiple readers, consequently increasing the cost.

2.1.4 Frequencies

RFID operates in different ranges based on the wireless communication and they form electromagnetic spectrums. The classifications of frequencies are generally divided into four bands. Other sub-bands also exist but their practice has not yet been fully exploited [4].

Each frequency band has a defined power limit that relates to the power output level of an RFID reader not the tag, due to a much higher power necessity. An RFID application is constrained to work within an authorized range that is associated with the allowed power source and frequency. However, the actual frequencies that RFID operates are within an unlicensed spectrum space and are limited to those frequencies, referred to as Industrial Scientific Medical (ISM). The regulations of these frequencies vary accordingly in different countries [1, 2, 4]. (Table was created from ref. [1, 3, 4])

Band	Frequency Range	RFID common frequencies	Reader range (Approximately)	Examples of applications
Low Frequency (LF)	30–300kHz	125–134 kHz	50 centimetres	1. Animal/Pet ID 2. Good for reading water content items at close range 3. Car immobiliser
High Frequency (HF)	3–300MHz	13.56 MHz	1.5 ~ 3 meters	1. Access and security control 2. Smart labels/cards 3. Library ID, 4. Clothing ID
Ultrahigh Frequency (UHF)	300 MHz–3GHz	433.05 MHz 866 – 954MHz 2.45 GHz	7~9 meters	1. Supply chain tracking: boxes, pallets, containers 2. Specialist animal tracking
Microwave	>3 GHz	2.4 – 2.5 GHz and 5.725 – 5.875 GHz	~30m	1. Vehicle fleet identification 2. Highway automated toll collection

Table 2: RFID operation frequencies

2.1.5 Standards

Countries apply limits on the power output, frequencies, and other operational constraints of RFID systems. Regulatory controls exist in order to specify the allowed strength and power levels according to the RFID applications carrier frequencies. (United States, Canada, Japan, Europe). The development of these standards is continuous and several bodies are responsible for simplifying and forming an enhanced functioning environment for RFID technologies in various regions around the world. Such bodies are: International Organization of Standardisation (ISO), European Telecommunications Standards Institute (ETSI), Federal Communications Commission (FCC), EPCglobal Inc. EPC global first approves RFID standards and then submits them to ISO, which approves them as worldwide standards [16].

The following areas require RFID standardization [4, 6]:

1. Air interface standards: Affects the RFID system components, the interrogator, and the tag, applying rules for communication between the two devices and other wireless communication systems.
2. Conformance: Tests that determine whether an RFID system fulfils its requirements against the standards.
3. Encoding and data structure: goods in supply chain travel to various locations and certain encoding and identification of both the product and details, such as the content, date of manufacturing etc, require the use of a standardised form and the organization of data.

4. Interoperability between RFID applications: use of technology for a particular purpose i.e. standards on portable labels.

2.1.6 RFID Applications

Constantly new areas are being introduced to the RFID technology. Some areas, such as management consider the technology as “the next big thing” due to the enhanced optimization of business process and the reduced operational costs. As the volume of tag manufacturing increases, the cost will consequently be reduced, thus proving further opportunities for the technology to develop into wider areas.

Over the past few years, RFID systems have been deployed in a number of application areas such as: supply chain management, asset tracking, healthcare applications, livestock and animal identification, human identification, retail, manufacturing, military applications and many other sectors. Below are brief descriptions of some of the most interesting examples that RFID applications are currently being deployed [2, 5].

Supply chain management: considered as the fastest growing area and the key driver for the development of the technology. Today thousands of companies across the world use smart labelling to tag cases, boxes, containers, pallets and other transportable carriers for products. There are incredible benefits for these companies e.g. monitoring their products, reading the entire content, reducing the time of process and preparation, instant identification of shipping containers in shipping yards with thousands of pallets belonging to other companies and so on [4, 5]. A recent example took place just after the catastrophic earthquake event in Haiti, in January 2010. The U.S. Department of Defense arranged to employ RFID to monitor shipping containers that contained supplies and aid due to the delayed responses because of the infrastructure loss and devastating state of the island [39].

Health Care applications: the adoption of RFID systems has begun in hospitals, were patients, doctors and expensive equipment can be tracked in real time. Active RFID are implanted into bracelets that allow patients requiring special attention to be continuously tracked. It enhances circumstances such as pre-surgery, supervising medication, blood transfusion etc. Bracelets are also applied to hospital staff to observe instant assessment. Access control is another benefit that permits access to critical zones of the hospital only to authorized personnel [28, 29]. Example: in the 4th annual RFID Journal Awards, The Roy and Patricia Disney Family Cancer Centre at the Providence Saint Joseph Medical Centre , was awarded for the most innovative use of RFID that reduced patient anxiety due to their idea to combine passive and active RFID systems for customizing hospital environments [33].

Human identification: there are many sectors that the tracking of humans involves RFID technology. E-passports and national identifications documents, schools and universities, prisons, museums, sports (race timing) [4]. However, these types of applications draw the most attention and debates against privacy issues. An example: China’s Jiangsu Longton Jail in the Nanjing Jiangsu province upgraded their security using TI-RFid 13.56 MHz ISO 15693 tags and readers. Around 6,000 prisoner wear wristbands that are used as an identification method, containing encrypted data including their name, ID number, and security level. The benefits of these systems are to establish a more secure environment, allowing accurate records of the prisoners’ whereabouts and management their personal information [34].

Other examples: Casinos are embedding RFID tags into chips so that the constant monitoring and tracking of the chips prevent theft, track gamblers habits and locate stolen chips e.g. Casinos in Macau are using 13.56 MHz RFID interrogators [35]. Libraries in the UK are also moving towards the technology, eliminating old style checkout systems using scanners that enable faster processing of the checking in and out of books, faster location on shelves and storing useful detailed information [36].

2.1.7 Future of RFID

It is a fact that throughout the literature research and review, the main point highlighted is that RFID has a promising future. Since the beginning of the decade up to now, the technology has touched many sectors and areas creating enhanced working environments with benefits that have not yet been entirely explored. Emerging RFID applications are essentially being developed daily and the technology will surely perceive groundbreaking progress with the support of large companies such as Wal-Mart, Proctor & Gamble, Metro, Target, Tesco, Air Canada, Fish & Richardson P.C. and other large enterprise companies [37].

According to other authors, the bigger picture lies in ubiquitous computing, otherwise known as Internet of Things (IoT). "Invisible", smarter, greater computational powers, cheaper etc, are all key trends devices must be capable of boosting in order to partake in completion of the pervasive computing vision. RFID is not long far from fully implementing these criterions and the technology already has the ability to identify individual items and their corresponding details. Everyday items embedded with such abilities and positioning them in numerous locations i.e. in our working and living environments will be able to connect to local networks that will then send their collected data to the Internet for further management. For instance, medical cabinets that automatically generate replenishment orders [38]. Hence, the commencement of the Internet of Things [6, 8, 20]. However, obstacles and issues remain prohibiting the overall success of the technology's deployment. Researches and RFID advocates are constantly labouring to overcome these issues and to provide solutions, interoperability, and harmonization. According to Lehpamer [4], the author reveals further technological issues, for instance environmental conditions i.e. temperature and humidity, management of data volume, spectrum congestion, availability of frequencies, reliability of the systems and other restraints discussed in further sections.

A study that took place in January 2007 by Vienhald and Wong [19], illustrated a number of important and unsolved issues that affect the future of RFID. The research employed experts to identify these obstacles and the likelihood of resolving them within the next few years. Twelve issues were found. Some of the topics are: the lack of RFID-skilled professionals, standardization, system cost, integration, privacy, security, and so on. Each issue was given a weight to provide sorting according to their importance. Nevertheless, it was noticed in the research that privacy was not a key factor yet the authors mentioned that the panel did not have a representative for the consumer prospective and that was the reason why privacy was ranked low. They further on mention that 'system cost' and privacy are linked and it is worth pointing out that cost was ranked second highest.

As the number of item tagging implementation will begin to grow rapidly in individual everyday items, so will the privacy concern arise and require serious consideration. Privacy and security are linked topics and as seen from the review mentioned above, cost ties in with privacy. Security remains an unsolved key obstacle for the future of the technology even though several studies are taking place at present.

2.2 RFID Security & Privacy

As mentioned earlier, the deployment and use of the RFID technology is growing rapidly across various industries. Even so, RFID raises serious privacy and security issues and in order to increase consumers' acceptance, RFID developers must assure that both the security and the privacy of data are entirely guaranteed.

There is no overall RFID security solution since RFID tags come in different flavours and require different security levels. As seen in table 1, all five classes of RFID tags explain briefly their functionalities and uses. Some low-cost basic tags cannot execute standard cryptographic operations like hashing, strong pseudorandom number generation, and encryption [13]. Other tags that are more expensive can afford to contain cryptographic algorithms but are usually not preferred in industries due to the affordance of a higher cost.

Privacy and security are strongly linked, taking into account that personal privacy and security of data depends on the type of RFID application. The degree of data sensitivity held on an RFID system requires a corresponding security level. A trade-off between cost and the consequence of privacy protection remains a major factor that forms a gap between RFID advocates and consumers. It must be stated though, that not all security problems are linked with human identification and privacy concerns. However, the concept of privacy opposed to security must not be misunderstood. Security threats *aim at* the RFID system, privacy threats are *formed by* the RFID system [27].

2.2.1 Properties of secure systems and the CIA Model

The CIA Model demonstrates a stepping-stone to endorse understanding and classify the areas threats and attacks can be found in RFID components and the system in general. When the security of an RFID is considered, a first approach is to take each dimension separately, C, I, A, and identify the threats and vulnerabilities that might affect the security and consequently the privacy of consumers. For example, when a system uses a central database such as EPC global Object Name Service (ONS) database, the detailed information of objects are tagged, stored, and managed. Confidentiality can be considered in situations where the data contains sensitive information e.g. medical care system of patients. It is a fact that the intervention of a sophisticated adversary or the leakage of protected access data by an insider will compromise security and the data can then be accessed over a global network or directly by the database. Hence, confidentiality of the data is violated and organizations or businesses will face serious consequences [1, 7, 17, 18].

There are other categories for instance complexity, usability and transparency that also define a secure system. A system's complexity must be at the minimum mandatory level, allowing easy maintenance and updates. A system's usability and transparency influence the relationship between a user and the system, if the operations are easy to understand and do not conceal any underlying operations [7]. Below is the diagram of the CIA Model and a brief description of each dimension.

Availability
Integrity
Confidentiality

Availability: The International Organization for Standardization (ISO), defines authenticity as "ensuring that authorized users have access to information and associated assets when required". An RFID system must ensure the availability, security, scalability and performance of the system when a user request a services such as the processing of data. An example of an attack is Denial of Service (DoS) through a blocker tag [1, 7].

Integrity: Ensures the prevention of unauthorized sources to modifying data and especially the sensitive of consumer information. ISO definition: "safeguarding the accuracy and completeness of information and processing methods". An example of an attack is spoofing tags that scan tags with an unauthorized reader to gain information or altering the original data upon transmission [1, 7].

Confidentiality: According to ISO, confidentiality is defined as "ensuring that information is accessible only to those authorized to have access". In other words, prevents the access of information to unauthorized personnel or systems. Access control and cryptosystems for securing data transmissions are ways to endorse confidentiality. Privacy and confidentiality are closely linked [1, 7].

2.2.2 Privacy

As long as companies, industries and other sectors that are already using or moving towards the deployment of RFID systems, do not take privacy issues seriously, these barriers will persevere, preventing the full adoption of the technology. RFID proponents are obliged to reassure consumers that their privacy will not be infringed, if they wish to gain trust and move into higher levels of RFID applications, such as human implants.

People can carry on themselves tags without their consent due to the small manufacturing sizes of tags. Activist are concerned about cases were readers are hidden in various locations and are able to detect an individual's personal items without them realizing and as a result monitoring a person's purchasing habits, whereabouts, personal possessions and daily patterns. Tracking is a highlighted topic in RFID privacy concerns, determining the individual's past and present location, without their knowledge. Tagging everyday items such as shoes makes it easier to reveal the course of a person and track their complete whereabouts, even outside their country. Other threats contain profiling, which is the leakage of information concerning an individual's possessions. Targeting also called the preference threat, that singles out a person according to their purchase preference can make them victims of direct marketing or other more serious risks such a scenario were a thief is interested in tracking an individual as a result of their specific purchasing patterns [15]..

2.2.3 Types of personal privacy threats

According to S. Garfinkel et al. [11], in the article the authors go on further categorizing personal privacy threats in more detail. Action, association, constellation, transaction and breadcrumb threats, all fall within the circumstances were RFID tags are embedded with unique ID numbers that can be linked to a person's identity. The above threats are closely connected to each other, and one may lead to another.

Constellation threat does not involve the identity of a person, yet an adversary can take advantage of a situation where the tags are associated with the person because they form a constellation, i.e. "aura". This can then lead to the transaction threat given that, the constellation of tagged items change location the transaction is simpler to gather linking the person with the corresponding constellation. These two types of threats can therefore indicate towards a breadcrumb threat. The word breadcrumb itself describes the type of threat. According to the author, breadcrumbs can be a number of tags a person has collected over a period of time that creates a record linked to their identity. Even with the disposal of the tags, the record still exist linking back to the original owner and if another individual uses those tags to cause a severe crime, the previous owner will be liable [11].

2.2.4 Examples of privacy threats in RFID applications

According to Lockton et al. [13], the authors present examples were RFID technology is used in three different areas that outline privacy concerns: item-level tagging, human implants and electronic national documentations e.g. e-passports.

Item-level tagging does not cause privacy threats when the use of the system solely ends at warehouses but as seen with other RFID application examples, consumers everyday items are already using tags to monitor information on products outside warehouses. An example the authors mention, is a case that occurred in the US, when Proctor & Gamble (P&G) tagged lipsticks with the intention of observing the reaction of customers when the items were taken from the selves, activating a camera that provided feedback to the P&G staff. In the UK, Gillette razors sold in supermarkets, worked in a similar way by activating hidden cameras to take pictures of customers and then comparing then with another camera activated at the store's exit. This illustrates a practical example of tracking people that consequently implies purchasing patterns, knowledge of their location, and even in extreme scenarios, theft when an unauthorized person scans the shopping basket [13].

Human implants is one the most sensitive sectors that privacy activist raise much concern and doubt in the deployment of RFID. Applied Digital Solutions (ADS) introduced a chip that is embedded in a glass container and implanted under the human fat tissue, providing automatic identification and information on the subject that is injected with the chip. This was first experimented in 2001 and since then many redesigns of the chip have been considered despite the fact that, ADS claimed that their system is secure. Global Positioning System (GPS) enhanced the design, allowing the identification to transmit its location through a mobile phone carried by the subject of the chip implantation. The benefits first considered were that the system would reduce and even prevent cases in countries with high ratios in kidnapping. However, the author states that this type of a system does not exclude tracking and security issues compared with other applications similar to item-level systems, which all the more makes it riskier to overcome these issues when the implantation requires *physical* removal. Activists insist that such systems must take into account the consent of subjects that are minors and clearly understand the benefits and drawbacks of such applications [13].

E-passports raise much debate in the security and privacy area such as surveillance, tracking, and even terrorist attacks. When firstly introduced, the technology of e-passports had to deal with several security issues such as skimming. (Passport information is read without the carrier’s knowledge). In April 2008, an article mentions Lukas Grunwald, a German security researcher and colleague Boris Wolf, that came up with a smart solution called RFDump and later on RF-Wall to prevent RFID fraud and attacks against e-passports, electronic access cards and e-payment cards using HMAC algorithm for the digital signature [30]. In the same year, another article, demonstrates a case were a fake passport of Elvis Presley passed undetected through security checks [31]. Other attacks, such as the one on the Oyster card for the London Underground, illustrates a Dutch hacker able to modify the information stored on the card and gain a free day travel [32].

According to a security analyst Paul Roberts [30], the real issue occur when companies and organizations are not willing to pay the cost for securing their systems. Systems that have already been widely distributed face great challenges in providing security and privacy, due to the numerous types of frequencies tags and readers work on.

All the above examples, illustrate that major projects of RFID systems still have to deal with privacy issues, which arise in the weakness of security. Although, the examples may show segments were privacy activist have amplified situations, it must not be forgotten that those activist not only play a major role in influencing others, they will not cease to exclaim their privacy rights as citizens until a solemn security solution is found. Ignoring these concerns will surely influence the growth and adoption of RFID technology.

2.2.5 Degree of relationship between RFID tags and human identification

Below is a graph that illustrates the relationship between tags in application examples and the ability to identify a person. (Original graph [17]). The degree of relationship axis is separated into sections, from none to absolute, that correspond to the level a person can be identified through an RFID tag. The RFID system type axis has three sections: restricted area e.g. warehouse or a prison, single organization e.g. bus company and multiply organization e.g. joint hospitals across Europe. Moving along the y-axis, the stored information on a tag spreads throughout cooperating companies or organisations. The same applies for the x-axis when moving onto higher levels, the vulnerability of information increases due to the stronger association of a person with a corresponding tag(s). For example, in animal identification no human privacy or tracking can occur. When considering public transport, purchasing an anonymous temporary ticket does not directly threat privacy. Nevertheless, constellation threat mentioned in section 2.2.3, states that eventually it can indirectly point to a person’s patterns. Payment methods are in the complete relation and multiple organisations because details move along the chain, from the store, bank, accountant and so on. Human chip implants are an absolute link between the person and the tag [17]. Applications for access control, such as the

Barcelona Baja Beach Club, insert tags allowing customers to access the club [40]. This cause serious privacy threats due to the low level of security in the specific tags [17].

Finally, the main point of the graph is to prove that privacy threats abide only in systems that form a partial or absolute link between an RFID tag and a person's unique identity.

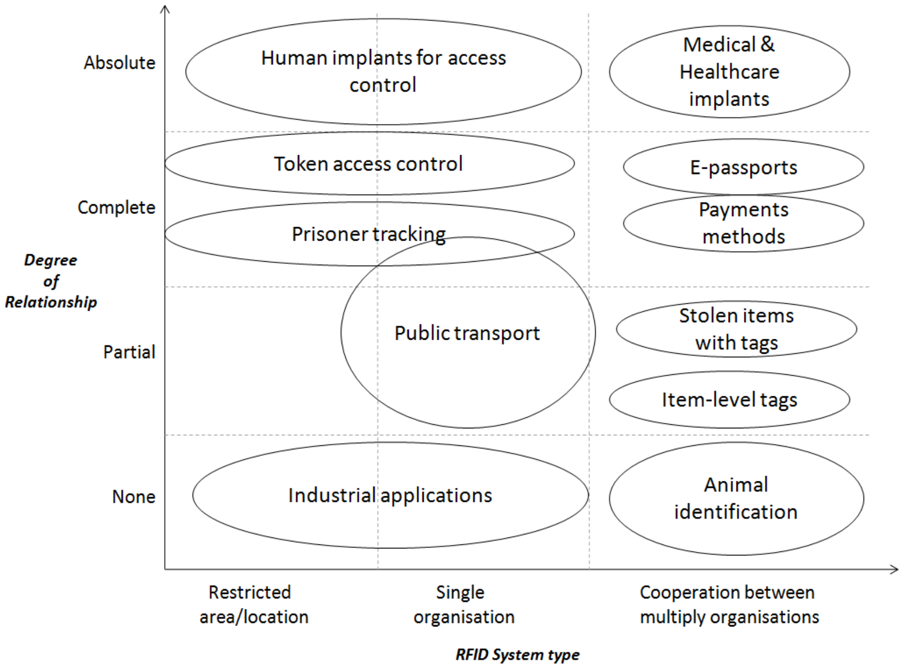


Figure 2: RFID tag vs. Human identification

2.2.6 Solving privacy issues through policy

A solution for RFID privacy and security issues can be through policy and self-regulations that governments or organisations can enforce on companies, industries etc to comply with the principles. There are various codes and Acts not only in the UK but also in Europe and the U.S that are been revised and employed into the world of fair personal information process and practice. Some of these codes are the Fair Information Practises (FIP), the "Directive on the Protection of Individuals" 1995 by the EU, the 1998 European Parliament guidelines on information privacy "European Community Directive on Data Protection" and others [12].

During literature research Dr Simson Garfinkel [11], was one of the names that appeared in RFID privacy literature. In 2002, Dr Garfinkel in the "RFID Bill of Rights" article suggested a framework for privacy policies based on FIP. There are five points the authors points out to consumers and users of RFID. The following points including a brief description [1, 13]:

1. If an item contains an RFID, the user has the right to know. E.g. representing RFID products with logos
2. When the use of the embedded tag is over, i.e. product purchased, deactivating, removing, or destroying must be available options to the user.
3. The information stored on an RFID tag must be available for the user to access and alter if necessary.
4. The reading of RFID tags embrace questions such as why, when and where, which should be available to the user's knowledge.
5. A user must have the choice to opt-out of RFID or use other alternatives to terminate the tag without losing the right to products and services. E.g., companies can use signs to inform users.

In the following year, 2003, Dr McCullagh went further on refining the privacy principles in a stronger manner, benefiting the end-user. The proposal consists that a tag must be deactivated by default after the point-of-sale and noticeable. No difficulty should be found in situations where a person wishes to remove a tag, and when feasible, tagging must take place on the package not the product itself [11]. Alternative technical suggestions by the Garfinkel are to include features on both the reader and the tag, such that distinct sounds or beaconing lights from either the tag or the reader can alert the time of interaction to the user. This brings other constrains when considering the tags cost, due to increase in the functionality [11].

2.3 Security

RFID applications that use tags of a higher cost can perform some type of cryptographic operations, such as shipping-container security, high-security contactless smartcards, and RFID-enabled passports [14]. However, basic tags i.e. passive, used in everyday balk quantities of items such as DVDs, clothes, soft drinks and so on, are pressured by industries and manufactures to keep the cost at minimal i.e. 5 cents per tag, and in the near future, pressure will force the cost to drop even lower, i.e. 1 cent [11]. According to the literature research, basic RFID tags at present, cannot successfully execute cryptographic operations e.g. encryption, hashing, strong pseudorandom number generation etc due to lack of resources [18].

This section will demonstrate the security threats found in various components of an RFID system. Subsections analyze findings from literature research in security and privacy surveys. First, the security diagram illustrates the security threats of an RFID system, moving onto a brief analysis on the types of attacks, technical approaches and finally the analysis of proposed cryptographic protocols and their limitations.

2.3.1 Security threats on RFID systems

The diagram below illustrates possible attacks and vulnerabilities in components of an RFID system. The air interface is considered the most sensitive area for attacks. If an adversary exposes one of these weaknesses and successfully proceeds with an attack, the RFID system can expose an abundant source of data [1]. (Original diagram [3])

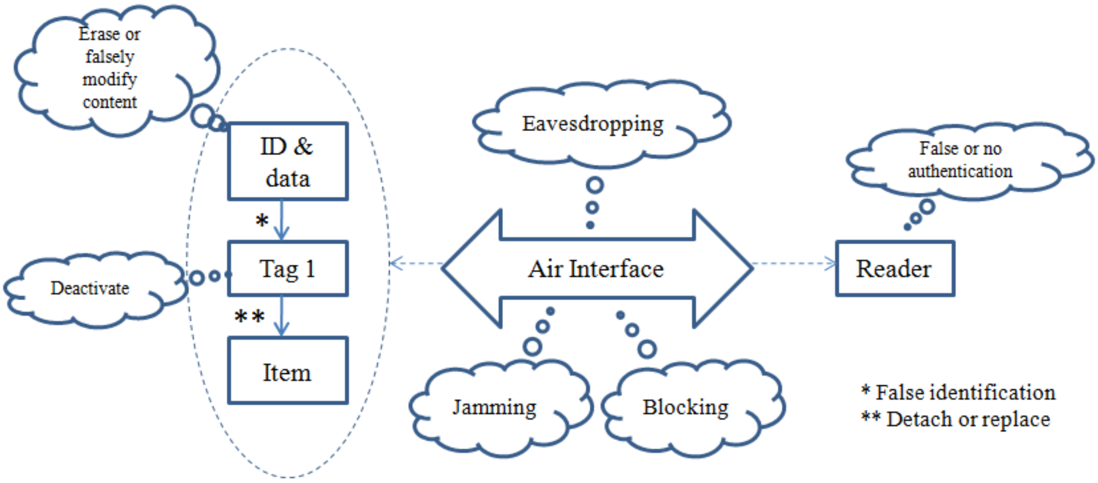


Figure 3: RFID system components and security threats

2.3.1.1 RFID Tag

The device itself is not complex so the basic functions it can perform are to listen and respond, no matter the source of the sender. Unprotected tags can face security threats such as eavesdropping, traffic analysis, spoofing, or DoS [1].

Weaknesses in tags [1]:
1. As seen in the diagram above, if a tag is visible it enables physical access, hence, vulnerable to physical removal or replacement.
2. Cost, complexity of process and additional functionality on the circuit, are all factors manufactures try to keep at minimal deployment owing to the low-cost tag demand.

2.3.1.2 RFID Reader

RFID readers communicate with the tags through the air interface, which is the communication channel between the reader and the tag. The reader also communicates through a wireless or wired Ethernet network to the middleware software that interacts with a central database. Some of the attacks are DoS, protocol attacks, spoofing, virus that affect RFID middleware etc [1, 27].

Weaknesses in readers [1]:
1. Data transmission from a tag to a reader is unencrypted.
2. The process of authentication must take place between a reader and a tag.
3. Physical location of readers must ensure some type of restriction to adversaries that are able to tamper with the readers or set up a system of malicious hidden readers for unauthorized reading of information.

2.3.2 Analysis of the types of attacks

Rotter in his article [17], states that the security of an RFID system involves a number of generic attacks. The type of attacks mentioned are eavesdropping, replay attacks, tag cloning, tag content manipulations, physical tag destruction or removal, blocking, jamming, and unauthorized tag reading. The author also states that when one attack occurs it is possible to directly or indirectly, initiate another type of attack. Below is a summary of the author's findings on the attacks, susceptible components, links between attacks and proposed solutions.

Eavesdropping: is a passive attack and hard to detect because no signal is emitted by the attacker. The attacker positions himself between the communication line of the tag and the reader, i.e. air interface, and tracks the data flow in the channel. Solutions are to encrypt the data and apply a suitable communication boundary for the distance between the tag and the reader. This attack can directly assist in tag cloning by copying the captured data and forming a duplicate tag. Indirectly affects tracking and replay attacks. Vulnerable components are tags and the air interface [17].

Replay attacks: this type of attack occurs when the communication between the tag and the reader flows through the established connection the attacker sets up. The attacker then gains information through the communication channel, granting him authentication that leads to further intrusion of the systems privileges. Authentication must not be compromised. Protocols such as the bounding protocol, which calculates the response time between the distance of the tag and reader, or challenge-response protocols are methods to achieve authentication. Other technical solutions could be the faraday cage or tags with a short range. Replay attacks indirectly lead to theft or damage to tagged items and supply the attacker with privileges such as entrance to constrained areas or spoofing access-control systems. Vulnerable components are the tags and the air interface [17].

Blocking: is the reverse of one of the technical privacy methods, meaning the same blocker tag used for preventing unauthorized readers from accessing the user's tags, now emits signals of numerous tags thus, generating a DoS attack.

Jamming: emits a radio frequency signal, i.e. noise; at the same frequency that an RFID system is operating on, as a result jamming the system. The jamming and blocker devices are easy to locate and developers have overcome these types of attacks with appropriate functionalities. Both jamming and blocking can directly cause theft or damage to tagged items and indirectly spoofing. Vulnerable component is the air interface [17].

Unauthorized tag reading: an unauthorized reader, possibly of an extended range can be used by an attacker to access and read information from a tag. The author recommends storing the information on a secure database, instead of the tag, and to limit the transmission time of the tag with manual techniques i.e. switches. The kill function on the EPCglobal Class 2 tags and physical damage are also possible solutions. This type of attack directly results to replay attacks, cloning, and tracking. Indirectly can cause tag content modifications, theft and item damage. Vulnerable components are tags, readers, and the air interface [17].

Tag content modifications: the attacker can modify the data damaging the functionality e.g. denial of access to a legitimate person, insert malware such as SQL injection that will later on affect the RFID's middleware software, distortion of item specifications such as price, date, etc. This applies only to writeable tags. EPCglobal class 2 Gen offers the "lock" and "permalock" function that can be a solution in such cases. In addition, the writing functionality design has the ability to allow changes to the memory only at specific times. This type of attack has a direct link with malware attack, spoofing, tagged item theft, and damage. Vulnerable component is the tag [17].

Tag cloning: an adversary can create a duplicate of a tag and then cause major damage depending on the tags functionalities. For example, changing item prices, purchasing with RFID-enabled payment systems or accessing constrained areas. Countermeasures could be reverse engineering and tag authentication via a challenge-response protocol. Replay attacks and spoofing can be direct consequences of cloning and can indirectly cause theft and damage to the status of items. Vulnerable components are tags and the air interface [17].

Another form of identifying security weakness is according to Thompson et al. [21], STRIDE model. Security threats in an RFID system can be grouped under the STRIDE security model that helps categorize attacks and design a secure system. Thompson in his RFID technical tutorial [27], briefly mentions methods that correspond to each category that possibly can overcome these threats. The six categories and their methods are [21]:

- **S**poofing identity: Spoofing is the case that an adversary masquerades as an authorized user.
  Method: appropriate authentication and avoid storing sensitive data.
- **T**ampering with data: tampering includes modifying, rearranging, deleting, or adding data.
  Methods: appropriate authentication, hashes, digital signatures, message authentication codes, and tamper resistant protocols.
- **R**epudiation: Repudiation occurs when an action takes place without the approval of the user and no proof is available to show that it actually happened.
  Methods: timestamps and digital signatures.
- **I**nformation disclosure: the visibly of data to unauthorized users.
  Methods: authorization, avoid storing sensitive information, encryption,
- **D**enial of Service: prevents authorized users for accessing services.
  Methods: Quality of Service (QoS), filtering, throttling, appropriate authentication, appropriate authorization.
- **E**levation of privilege: an unauthorized user gains legitimate rights that elevate his privileges in a system.
  Methods: limit access rights where not applicable.

2.3.3 Technical methods to RFID privacy

There are several technical solutions for solving privacy issues. Below are the descriptions and limitations of some of the most popular technical methods found during relevant literature research. These methods are mostly seen in basic tags that cannot perform cryptographic operations.

- Kill function: item-level tagging applications particularly employ 'tag killing'. The function is a built in option the EPCglobal standards proposed for the Class 1 Generation 2 UHF Air Interface Protocol [4]. The 'killing' of the tag occurs once the consumer ends their purchase, to ensure that the tag cannot be detected any further. The benefits for using such a function are low cost, easy to understand and accepted by consumers.
  Limitations: Restricts commercial users and stakeholders to take full advantage of RFID-enabled "smart" applications and other future potential emerging services. For example, when returning or repairing products, locating high value stolen items and even "smart" microwaves or fridges that automatically follow instructions from tags [11, 13]. Areas that the kill tag functionality is restrained are indoor tracking because RFID enabled shelves can activate hidden cameras before the point-of-sale, as seen in an example in section 2.2.4.
  Alternatives can be to use temporary kill functions that use software lock or other mechanisms to allow the tag to 'sleep' and then be re-activated in the future, with or without the consumers consent. Overwriting data with zeros is another option yet remains vulnerable to tracking because only certain stores will use such a system [13].
- Blocker Tag: RSA laboratories developed a passive RFID blocker tag that uses sophisticated algorithms to form a zone of privacy around the user that prevents unauthorized scanning. These algorithms allow the tag to broadcast all possible RFID identification codes overloading the reader's anti-collision protocol and simultaneously blocking the functionality of a reader [18, 22]. Anti-collision protocol e.g. tree-walking protocol prevents a broadcast collision when multiple tags simultaneously transmit signals to a reader. The tree-walking protocol can be described as a tree which nodes represent a prefix identification code and the left child corresponds to 0, the right to 1. According to the broadcast code of the tag i.e. 1 or 0, the reader recurses either or the left side of the tree or the right, respectively. A blocker tag is constructed of two antennas that reflect back the two bits simultaneously; thereby the reader attempts to search the complete tree forming a type of passive jamming [11].
  Soft blocking is an alternative to the blocker tag in the sense that it relates to the "polite" policy, meaning it simply notifies the readers without flooding them with simultaneous series of responses. The reader must also apply to the polite policy and when scanning the tags, if it discovers a particular tag identifier corresponding to the blocker functionality, it must ignore the privacy zone of that specific tag [11].
  Limitations: A malicious blocker tag can cause damage in an environment that uses an RFID system by generating a Denial of Services (DoS) and adversaries can simulate readers to overcome the functionality of a blocker tag [11]. Orientation is another limiting factor [18]. Consequently, the blocker tag is not the ultimate solution in situations such as airports that use electronic identification or hospitals using implanted RFID-tags for patients, requiring the blocker tag to be temporarily disabled to avoid disruption of the systems [13].
- Antenna power analysis: According to the Garfield [11], analysing the antennas signal was proposed by Fishkin and Roy, in scenarios where malicious readers are likely to be further way than the applicable reader. Adding functionality to the circuit of an RFID tag enhances it to measure the distance of querying readers, thus identifying the malicious reader and react as necessary. This type of scheme can be merged with other techniques such as the challenge-response protocol and the blocker tag [4].
  Limitations: basic tags lack the ability to precisely measure distances and a standard for defining the ultimate minimum required distance is different in varying systems [18].
- Faraday cage or aluminium foil: a practical approach that restricts tags from being read when shielded in aluminium foil or a metal screen. It is a cheap and easy method that passports or wallets carrying tagged cash can overcome privacy issues by applying a metal screen to block

malicious readers from accessing the information [13, 16]. Companies are selling aluminium lined wallets and purses to block radio signals from reading passports and RFID credit cards [18]. Limitations: mainly not considered the ideal solution to wrap items in metal mesh due to obvious reasons and even so items such as clothes and implants cannot be covered by a foil-line cover. [13, 18]

- Tag passwords: this approach is currently under revision by researchers due to the limitation of password-enabled tags in multiple environments requiring password management. EPC tags can be equipped with passwords and useful in specific systems that implement readers capable of sending the correct password and releasing the tags information [11].
- Other methods: Vindictive Sentinel device, also referred to, as the Agent Scheme is a device that requires the legitimate readers to be registered whereas the rest of the readers (unregistered) are blocked [18]. The RFID Guardian lets users upload lists that define access control rights to specific parties and operations. This method also randomly alters the jamming radio signal [18]. Another method suggested by Karjoth and Moskowitz[14, 18] is to allow the user to physically separate the antenna form the tags restricting the reading of the tag to centimetres but is limited only to external tags. Inuou and Yasuura [18], suggested that a tag is constructed in two parts and the part that contains the identification can be separated at the point-of-sale, whereas the rest of the details and data remain.

It can be considered that constellation threats remain in all the above techniques, if tags carry a specific technical type for privacy protection, forming a unique constellation of the multiple tags a user might be in possession of.

2.3.4 Other security methods for basic tags

There are a number of proposed security and privacy methods for basic RFID systems mentioned in the article by Juels [14]. The author mentions that the lack of cryptography in passive RFID tags has been a challenge to researchers. Lightweight approaches such as the "minimalistic" cryptography proposed by Juels [76], works as follows, a legitimate reader stores the set of pseudonyms that corresponds to a specific tag. Then the tag recurses through these pseudonyms and emits a different one at each query from the reader. An unauthorized reader will not be able to interact with the tag, because of the lack of knowledge of these sets. Another method mentioned by the author, is re-encryption by Juels and Pappu [14]. One the one hand, it is proven to be a heavy system when applying re-encrypt to readers and vulnerable to eavesdropping. On the other hand, this solution overcomes situations for basic tags that are limited to cryptographic functionalities. Universal re-encryption is an extended version of the re-encryption scheme [14]. Instead of using a single key pair, secret key (SK) and public key (PK), this method introduces multiple key pairs (SK_i, PK_i). A serious limitation occurs at the re-encryption of a ciphertext that requires the corresponding PK_i, revealing sensitive privacy information. However, Golle et al. [14], introduced the El Gamal cryptosystem that resolves this limitation. Ateniese et al. [14], later on improved the solution with elliptic curve cryptosystem but the new approach is weak against swapping attacks. Chapter 3 investigates twelve protocols similar to the ones discussed in this section and chapter 4 reveals their weaknesses.

Although, numerous solutions might seem available to address security and privacy concerns, all the proposed solutions up to now have some form of limitation or neglect to cover important features [14]. Especially concerning basic tags, were most of the research has been directed towards active tags or more powerful computational devices and another factor is the commercial pressure that grows at present that tries to keep the cost low [11]. Langheinrich in his article [18], discusses similar findings and highlights issues such as pseudonym updates, ownership transfer and key management that cryptographic primitives fail to resolve in low-power and cheap RFID components. The following section describes cryptographic primitives that are frequently used in RFID protocols.

2.3.5 Cryptographic Primitives used in RFID Protocols

This section provides a comprehensive overview of cryptography primitives that forms the basic background for understanding the following chapters that analyze RFID protocols. According to Jantscher et al. [64], cryptographic primitives can be separated into unkeyed, secret-key and public-key modern cryptographic primitives. The functionalities, performance, security level and implementation complexity can vary across different primitives. Below is a diagram of the cryptographic primitives with examples for each category and an accompanying description. (Original diagram and following explanations [64])

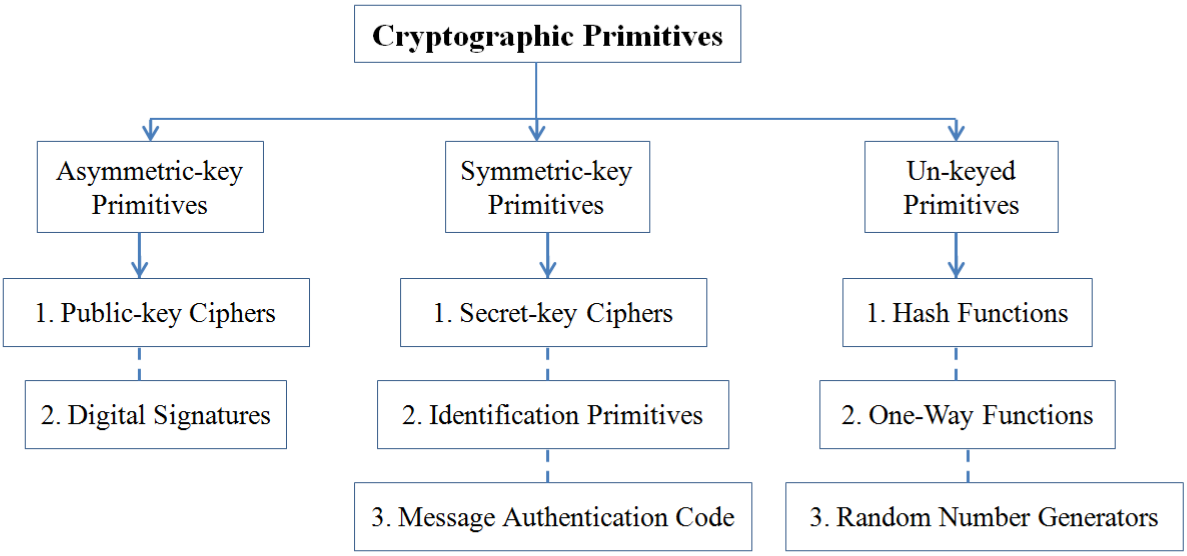


Figure 4: Examples of Cryptographic Primitives

1. Unkeyed primitives: primitives that do not include the use of keys and are usually combined or implemented with other operations in order to provide sufficient security. These primitives are often seen in RFID protocols because they are cheap and generally fast if there usage is kept to a reasonable number. Examples are:

   1.1 Hash-functions: should be a one-way function meaning, given a hash value of a message m, it must be computationally infeasible to calculate m=h(m) from h(m). Another essential property is collision resistance i.e. given two message, m_1 and m_2, they must not hash to the same value i.e. $h(m_1)=h(m_2)$. Finally, hash-functions must include randomness. In other words unable to distinguish a random mapping from hash function. There are two types of attacks, pre-image and collision attacks.

   1.2 Random number generators: are independent and uniformly distributed ensuring equal probability in generating a value from a group of numbers and the next or previous value to be generated does not depend on the most recent one. Random numbers generators are frequently found in RFID cryptographic protocols.

   1.3 One-way functions: there exists a polynomial algorithm that can easily map a value e.g. f: r →R. Informally, they are very important primitives also frequently used in RFID protocols that map values in such a way, that they are easy to compute but hard to invert.

2. Symmetric-key primitives: are secret-key primitives that share common secret keys between trusted parties, e.g. secret key stored in both the reader and the tag used for authentication purposes. Examples:

   2.1 Identification primitives: Also known as, entity authentication is a technique used when one party wants to ensure the identification of another party aiming to prevent impersonation attacks. Generally, there are three techniques an identifying party can utilize, something known i.e. password, something possessed i.e. smart card or something inherited i.e. fingerprints. RFID protocols use challenge-response techniques that generate a random nonce usually from a random number generator to prevent replay attacks and achieve strong authentication in contrast to fixed passwords proven to be a weak form of authentication.

2.2 Message Authentication Codes (MAC): are similar to hash functions with the addition of key usage. MACs are used to prevent adversaries from tampering with data and are able to identify any modification of messages while transmitted through a communication channel. These functions essentially provide integrity of data.

2.3 Secret-key ciphers: Symmetric-key cryptography can encrypt and decrypt messages using a secret key. There are two types, block cipher and the stream cipher. Block ciphers separated the message into blocks of the same length whereas, stream ciphers use a pseudorandom cipher bit stream most common is an XOR operation in combination with the encryption of each digit of the plaintext. Block cipher e.g. DES, AES, Stream ciphers e.g. LFSR.

3. Asymmetric-key primitives (Public key cryptography PKC): are based on public/private keys were the former is used for the encryption of information and the later for decryption. PKC is one of the largest groups of cryptographic tools and can be combined with symmetric-key primitives to provide additional security.

3.1 Asymmetric-key ciphers: Party A has possession of the public key PK, which is also visible to un-trusted parties and party B stores the secret key SK. The message m is encrypted with PK and the ciphertext C is then sent to party B that can obtain the decrypted information with SK. An issue that can arise from PKC is the encryption of messages from illegitimate parties that can eventually determine the SK for decryption. Hence, a practical solution is public-key-infrastructure (PKI) that helps prevent such attacks using certificates. Examples are RSA cryptosystem and Elliptic Curve Cryptography (ECC).

3.2 Digital signatures: is a technique that combines the identity of the entity with the corresponding message. The signing entity i.e. Prover is in possession of secret key (sk) and there also exists the verification key (vk) used by the Verifier. The Prover signs the message with sk and produces σ which is then sent to the Verifier. The Verifier has vk, σ and has to either output 1 or 0 i.e. verify/reject, respectively. Example is the ElGamal digital signature scheme.

3. Description of Several RFID Protocols

This section consists of the analysis and description of several RFID protocols. Inclusion criteria for choosing the protocols are based upon the popularity of the paper, recognition of authors, undertaken research by universities and diversity of techniques used. The analysis of both weak and strong RFID schemes, structure a wider investigation in the current state-of-the-art, allowing the assortment of constructive information for the later phase of the project.

Protocols are number throughout this chapter, at the beginning of each description. Protocol numbering i.e. Protocol 5, will be used as an abbreviation in chapter 4.

3.1 EC-RAC (ECDLP Based Randomized Access Control): Provably Secure RFID authentication protocol (version I)

Protocol 1 [65]. The authors in this paper propose the first version of the EC-RAC protocol for low-cost RFID. Security of RFID tags can sometimes be compromised in order to sustain low-cost. The authors claim to deal with this issue by designing the protocol based on Elliptic Curve cryptography that involves small key size and low computational requirements. The design of the protocol is based on public-key cryptography instead of symmetric-key with the aim of satisfying three fundamental security properties all RFID systems must meet; scalability, anonymity and anti-cloning. However, authentication protocols that are based on public-key cryptography, as described by the authors, also lack the requirement of anonymity that can lead to tracking. EC-RAC claims to be a secure protocol that overcomes the above issues and uses minimal computational power. Below is a brief description of how the protocol works and the reasons for choosing the basic design fundamentals [65].

The authors mention two types of secret-key cryptography algorithms, fixed access control, and randomized access control. The difference between the two designs is the tags response to the reader that can either include a fixed or randomized message. The two major drawbacks are tracking and scalability, respectively. EC-RAC uses randomized access control and deals with the issue of scalability by keeping the computation workload minimum for the tags. This is achieved by transferring the major workload unto the reader/server. As a result, the protocol avoids the workload increasing linearly, as the number of tags increase. An important point in this protocol, are the two point scalar multiplication for the tags that reduce the workload. Other authentication protocols mentioned in the paper include the Schnorr and Okamoto schemes based on ECDLP (Elliptic Curve discrete Logarithmic Problem). The authors eliminate these designs since tag anonymity is not considered and the tag's ID, which corresponds to the public key, is public, overseeing the requirement that the tag's ID must be kept secret when being transmitted that leads to tracking as well. EC-RAC uses public-key cryptographic and an ECC based algorithm that minimizes the computational workload of the tags and makes sure the public key, which is the tag's ID, is kept secret [65].

The protocol applies an elliptic curve over a finite field. Publicly known points are: P, $Y{=}yP$, x_1P, x_2P, Scalar: y (stored on the reader), scalars: x_1, x_2 (unique and stored on each tag). The authentication process starts by generating two random numbers, r_1 and r_2, and the reader sends r_2 to the tag. The two secret keys are X_1 and X_2 that correspond to the ID and password. Public keys are x_1P and x_2P that are used as the ID and password verifiers[3]. The difference between other protocols based on public-key cryptography is that public keys are public to anyone that has the ability to tap into the communication channel, whereas in EC-RAC stores the keys securely in the server in order to prevent tracking and to transmit the tags ID securely. The tag then checks if $r_2 = 0$ and if so aborts, otherwise three messages are generated by the tag and send to the server. The server then executes a number of

[3] Note: the authors state that the public keys $X_1 = x_1P$, $X_2 = x_2P$ are kept secret.

calculation to obtain X_1 to will be used to find x_1 and X_2. If a match is found the server then recalculates and compares the results with the stored X_2 and accepts the valid tag otherwise rejects. Below is a diagram of the protocol (Original diagram [23, 65]).

Tag (Prover)	Reader (Verifier)
x_1, x_2 (Secret keys), $Y \leftarrow yP$	$X_1 \leftarrow x_1P$, $X_2 \leftarrow x_2P$ (Public key), y, x_1

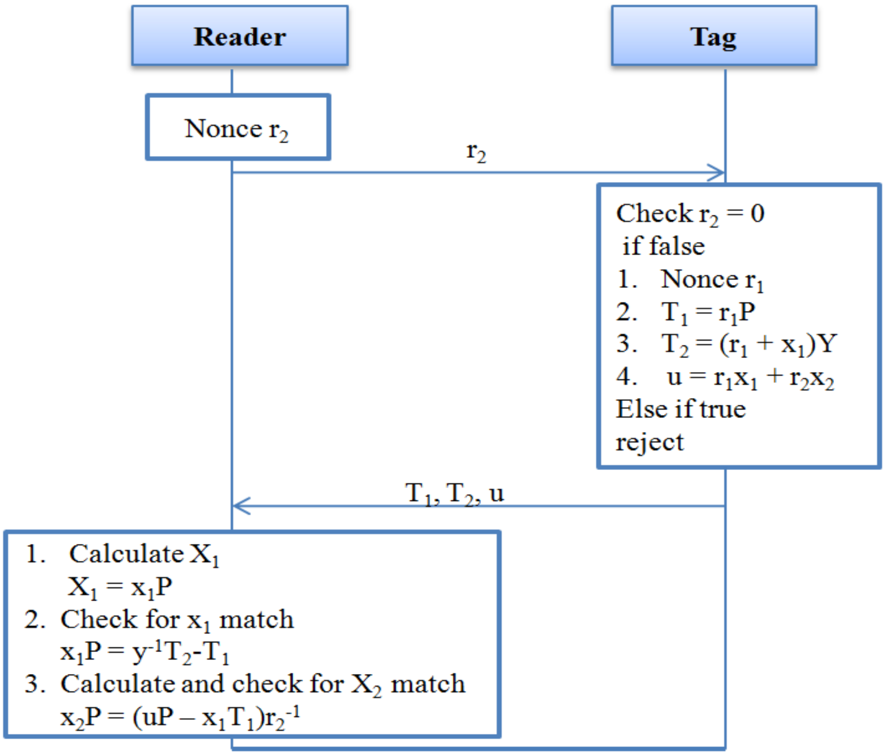


Figure 5: EC-RAC (Version 1) Protocol Diagram

The authors claim that EC-RAC is secure in the generic group model. According to the authors, this model conceals the actual group elements and allows only the images of the elements that are mapped to random strings to be accessible by the attacker. Other proofs are also used to prove the security of the protocol i.e. third observer attacks, illegitimate tag impersonation attacks, illegitimate reader impersonation attacks, and tracking attacks. Finally, the protocol achieves security in the generic group model, minimizes computational requirements, compatible to other authentication applications, scalable, secure against cloning and tracking and uses public-crypto algorithms that enables the secure transfer the tag's ID.

3.2 Untraceable RFID Authentication Protocols: Revision of EC-RAC

Protocol 2 [66].The previous version of EC-RAC was proven unsecure and according to Deursen et. al.[81], Bringerl et.al [88] the protocol lacks certain security properties such as un-traceability. Detailed description of attacks are described in chapter 4. The authors in EC-RAC version I, expanded and revised the protocol by proposing several authentication protocols constructed by the same blocks. They claim to satisfy all RFID system requirements including extra security properties such as replay attacks and backwards/forward un-traceability that were not considered in the first version. Below is a brief description on the protocols designs and their security properties. The components that form the authentication protocols are:

1. ID- transfer scheme: the ID –verifier of a tag is encrypted and transferred to the server that is responsible for decryption.
2. Password transfer scheme: the password-verifier is encrypted and transferred to the server, which will then decrypt and authenticate the tag.
3. Server authentication: involves a session identifier sent to the tag, allowing the tag to verify the server.

Six different authentication protocols are designed according to the various combinations of the above blocks, which also vary in their computational power demand. Each of these protocols commences differently to the requirements and security properties various systems might have. The authors claim that the security properties of the protocols are proven using reduction[4], meaning that the proof of security of these protocols lay within the level of security of the underlying primitive schemes. The security analysis of the protocols are described for each of the three parts independently, assuming that if all schemes are secure then the combination that forms the six authentication protocols achieve overall secure designs.

Cryptographic reduction of the Schnorr protocol and Diffie-Hellman scheme is used as a means to prove the three schemes are secure. For example, the ID transfer scheme is shown to be secure against replay attacks, by reducing it to the Schnorr protocol, which is proven to be secure against replay attacks. However, Schnorr is not secure against tracking attacks, so the authors reduce the scheme to the Diffie-Hellman scheme in order to prove security against tracking attacks. The security analysis of the password transfer scheme is presented in a similar manner and includes the same security properties. The security analysis of the server authentication process is not emphasized in the paper since servers are not required to repeatedly authenticate themselves to tags. Hence, a session identifier is sent from the server to the tag at the setup phase of the system, in order to prove its authenticity and requires no other information to be encrypted or decrypted. Below is a table that illustrates the security and performance analysis of the six authentication protocols (Original table [66]).

Components	Protocol 1	Protocol 2	Protocol 3	Protocol 4	Protocol 5	Protocol 6
ID-Transfer	✓	✓	✓	✓	✓	✓
Psw-Transfer 1	X	✓	X	X	✓	X
Psw-Transfer 2	X	X	✓	X	X	✓
Server Authent.	X	X	X	✓	✓	✓

Figure 6: Six Authentication Protocols with Different Combination of Components

According to the authors, there are basically three strong points about their proposals. Firstly, an RFID systems with varying requirements and security properties, can inherit one of the six protocols with the most efficient combination of schemes. Secondly, the computational workload of tags are kept to a minimum. Third, the security of the protocols are proven with cryptographic reductions of Schnorr protocol and Diffie Hellman, two well know and efficient techniques to prove the security of protocols.

3.3 Low-Cost Authentication Protocols for RFID

Protocol 3 [67].According to Lee et al. [67], the proposed protocol is the third version of EC-RAC that was extended and corrected due to the vulnerabilities and limitations of the previous designs. The ID-Transfer scheme was proven vulnerable to a man-in-the-middle attack. The authors mention cryptographic hash function as a possible solution to the man-in-the-middle attack but eliminate it due to the limited die-size of tags. Instead, they persist with EC-operations with non-linearity. The authors claim that the two new RFID authentication protocols are narrow-strong and wide-weak privacy-preserving. As with the previous versions, the two protocols differ in the computational workload and

[4] Reducing a protocol means that the attacker has more adversarial power or information [66].

security properties. The present a new scheme, “the search protocol” that allows a specific tag to be located by a server without reducing privacy.

In this version, privacy is emphasized stating that the level of resistance an RFID system has against tracking and anonymity reflects on the effectiveness of protocol as well as the level of security. Vaudenay’s theoretical framework and his terminology [56], is used as a reference to map the privacy and security properties of the protocols. The wide and narrow attack occurs when the results of a protocol i.e. accept or reject a tag, are either exposed or hidden to the attacker. If the results are hidden, the attack is characterized as narrow. A strong attack occurs when the secret of a tag is no longer kept secret and can be reused by an adversary. Otherwise, it is a weak attack. Hence, a wide-strong attack is considered the strongest form of attack. The authors claim that their new proposed protocols offer narrow-strong and wide-weak privacy.

The security proof of the ID–transfer scheme is presented in a similar manner as previously. The authors this time analyse the schemes into two parts; security and the privacy. In order to illustrate the security proof, the scheme is reduced to both the Schnorr protocol and the Decisional Diffie-Hellman problem to demonstrate the security against replay and tracking attacks respectively. Privacy analysis focuses on proving narrow-strong privacy and the efficiency of preventing a wide-weak attack. The ID and the password are represented by two secrets, x_1 and x_2 which are independent of each other. The Psw-Transfer scheme is considered equal to the ID-Transfer scheme and both achieve the same privacy and security properties. The ID-Transfer scheme can be used alone as an authentication protocol. However, when both schemes are combined, they accomplish additional security properties but require larger amounts of stored information.

The diagram below illustrates the combined ID&Psw-Transfer scheme. The authentication procedure between the tag (prover) and the server (verifier) initializes with the tag generating two random numbers nt_1 and nt_2. These two nonces are used to compute T_1, T_2 that are sent to the server. The server chooses a random number nr_1 which donates the x-coordinate of x_1P and sends nr_1 to the tag. The tag then checks to see if it matches the x-coordinate of nr_1P and if so computes T_3, T_4 and sends them to the server. The server’s final step is to calculate and find x_1 and X_2 and if they match the stored values then it accepts the tag as a valid one otherwise it rejects. (Original diagram [79]).

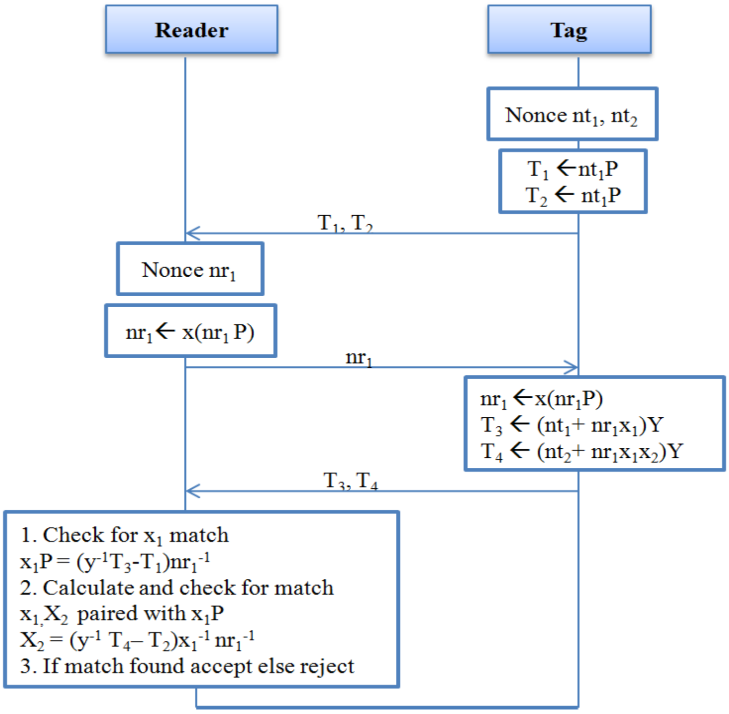

Reader
Tag
Nonce nt_1, nt_2
$T_1 \leftarrow nt_1P$
$T_2 \leftarrow nt_1P$
T_1, T_2
Nonce nr_1
$nr_1 \leftarrow x(nr_1 P)$
nr_1
$nr_1 \leftarrow x(nr_1P)$
$T_3 \leftarrow (nt_1 + nr_1x_1)Y$
$T_4 \leftarrow (nt_2 + nr_1x_1x_2)Y$
T_3, T_4
1. Check for x_1 match
$x_1P = (y^{-1}T_3 - T_1)nr_1^{-1}$
2. Calculate and check for match
x_1, X_2 paired with x_1P
$X_2 = (y^{-1} T_4 – T_2)x_1^{-1} nr_1^{-1}$
3. If match found accept else reject

Figure 7: EC-RAC (Version 3) Protocol Diagram

The design of the search protocol enhances the search and location of a specific tag from many. The authors claim that the computational workload of the previous protocols increases linearly as the number of tags increase in the system whereas, the search protocol is scalable and prevents tracking attacks. In addition, the search protocol is a one-round authentication, meaning it only take one round for the protocol to complete with a single message and only the targeted tag is able to verify the message sent by the server.

This version of EC-RAC draw is proven once again vulnerable to several attacks by other authors. The authors claim that Protocol 3, which is the combination of the two schemes achieves a higher level of security. The search protocol can be demonstrated as a practical protocol for example in library applications were the prompt search of a specific tag book is requested. However, the authors themselves state that the search protocol is not narrow-strong privacy preserving, allowing an attacker to extract useful information from the tags.

3.4 Low-Cost and Strong-Security RFID Authentication Protocol

Protocol 4 [68].The authors of this protocol use previous protocols as a guide to design their new authentication protocol. Two interesting points of this design, is the attempt to use standard techniques and hash functions that result in limited computational requirements. Their main aim is to reduce the computational workload on the back-end database and if a state of desynchronization occurs, the protocol is supposedly capable of re-synchronizing the tag and the database in the following session. The authors also claim the protocol secure against replay attacks, spoofing attacks and provide forward secrecy. Below is a brief description of the protocol. The original table, diagram and protocol explanation can be found in [68].

Terminology
ID: identity of tag
HID: hashed value of the ID stored in the database
PID: Previous ID of the tag from previous session
X_T, X_R: random numbers generated by either the tag or the reader
S: Synchronization State

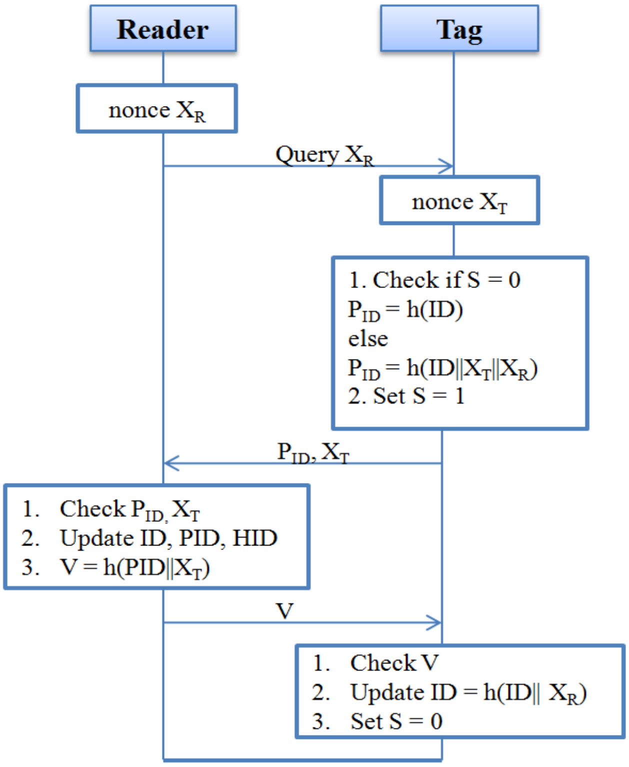


Figure 8: Protocol 4 Diagram and Terminology

The first step starts when the reader query's the tag with a random number X_R. The tag then generates a random number X_T and according to the state S it computes P_{ID}. If S is 0 then P_{ID} is equal to the hashed value of the tag's ID i.e. *h(ID)* else the tag sets S to 1 and P_{ID} =h(ID|| X_T|| X_R). P_{ID} and X_T are then sent to the reader, which forwards the received information and its own X_R to the database. The database handles most of the workload and executes four calculations. First, once it has received P_{ID} then it tries to match the specific tag by comparing the hashed ID, i.e. *h(ID)* received by the reader with the stored hashed ID's. Second, if a match is not found then P_{ID} is compared with $h(ID||X_T|| X_R)$. This occurs in situations when the S is set at 1 but messages have been blocked during the previous sessions resulting in updating desynchronization of both the tags ID's and the database's stored ID. Third, if the second case cannot find a match the database calculates the *PID* values with the following, $h(PID||X_T|| X_R)$ and compares then with P_{ID}. This situation occurs when the ID in the database have been update but not the tag's current ID. The procedure will halt if in an all three searches no match was found. In the final step, the database computes $V = h(PID||X_T)$, then updates *ID*, *H(ID)* and send *V* to the reader which will forward it the tag. The tag then calculates *V* and if it is correct, the tag updates the ID as well.

The authors demonstrate a basic security analysis that includes proof of security against desynchronization attacks, location attacks, forward security, spoofing attacks etc. The following chapter, contradicts these claims with attacks. Below are points the authors mention about the security of the protocol:

- The one-way property of the hash function prevents an attacker from obtaining any secret information or the ID.
- Spoofing attack is prevented since an attacker must be able to compute P_{ID}, which is unfeasible without the *ID*. An attacker is unable to impersonate a reader as well, since calculating V requires the knowledge of the ID.
- Replay attacks are prevented using the random numbers X_T, X_R.
- Location tracing: the attacker is unable to mount this attack due to the refreshing of the ID at for every session. Indistinguishability of tags is also a requirement to prevent this type of attack. The authors claim that when the ID, using a one-way hash function in the previous session is renewed, the attacker cannot distinguish two different tags.
- Forward security is preserved since the attacker must have access to the tag's ID, which according to the previous statements it is impossible. Therefore, locating a tag backwards is also prevented. However, continuous desynchronization allows an attacker to collect all the messages transmitted during the protocol run that will eventually reveal the tag's ID. Forward security in such circumstances is hard to satisfy and the proposed protocol achieves this property only for successful ID updates.
- Desynchronization attack: there are generally two phases a desynchronization can occur. First, when the attacker blocks the response message emitted from the tag. Searching for the correct ID stored in the database can re-synchronize the system. Secondly, if V is blocked from the reader to the tag, the database and tag can search and find the stored PID, which will restore synchronization.

3.5 Secure and Low-cost RFID Authentication Protocols

Protocol 5 [69].Lee et al. [69], propose two authentication protocol, Semi Randomized Access Control (SRAC) and Advanced SRAC (ASRAC). SRAC is based on hash functions whereas ASRAC uses a randomized number generator. The authors provide an explanation for the operational requirements such as scalability and security requirements such as tracking problem, cloning attack, replay attacks, DoS and forward secrecy, and demonstrate how the proposed protocols meet these requirements. Although, both protocols designs are interesting, the authors hint towards a vague statement that SRAC and ASRAC are weak against tracking attacks.

In the SRAC protocol, the reader query's a tag and the tag then responds with the hashed ID, *h(ID)*. The server then checks to find a match of the current ID in the database and generates a random number N_R used with an XOR operation to produce the new ID, i.e. $h(key \oplus N_R)$. The server finally checks if new ID is unique in the database and updates accordingly.

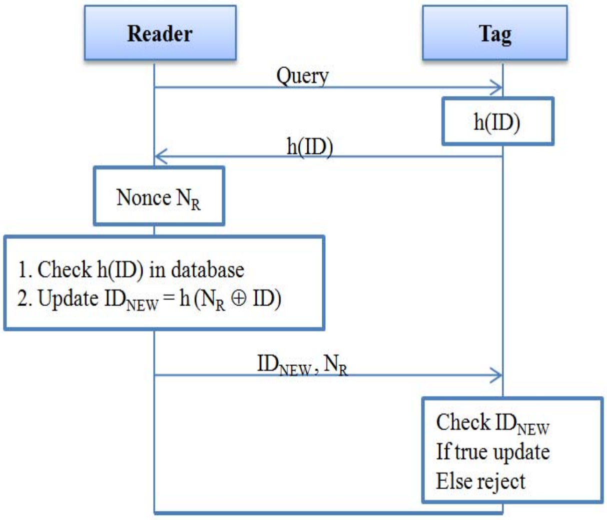


Figure 9: SRAC Protocol Diagram

A-SRAC protocol is designed analogous to the previous one. There is an extra feature that prevents replay attacks, using a random number generator from both the tag and the server. A mutual authentication process takes place between tags and servers, which generate challenges and expect responses. The server sends N_{R1} to the tag which in return replies with *h(ID)*, $N_{T,}$ and $h(key||N_{R1})$. Once *h(ID)* is received by the server, it then authenticates tags by checking $h(key||N_{R1})$. Then a second random number N_{R2} is generated and computes the tag's new ID, i.e. $h(ID \oplus N_{R2})$. Tags authenticate servers by checking if the $h(key||N_{R2}||N_T)$ is correct and update the ID's accordingly.

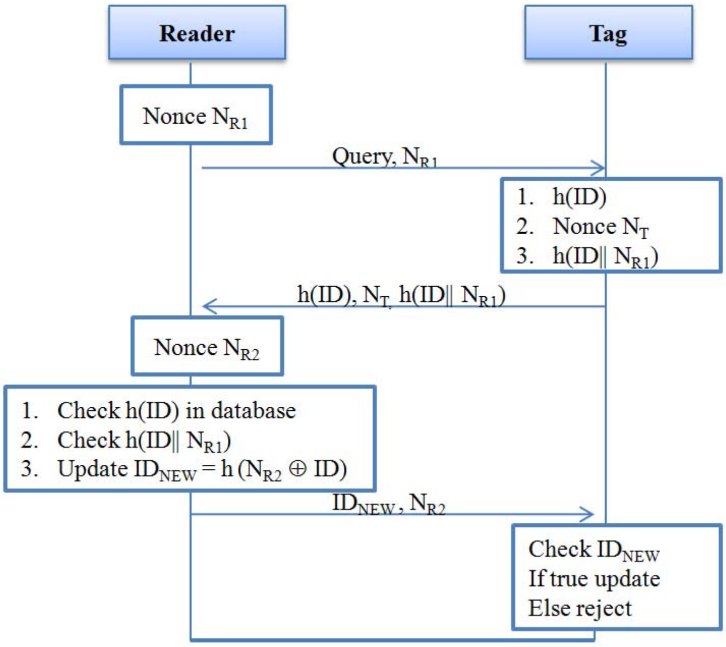


Figure 10: A-SRAC Protocol Diagram

Important notes mentioned in the paper that are useful when considering the design of an RFID protocol:

1. Similarly with other protocols, the authors assume that the communication channel between the reader and the server is secure.
2. Most energy consuming phase in an RFID system is the amount of tag transmissions. By reducing the transmission, then the overall computation workload is also reduced.
3. The conflict of similar tag ID's can cause serious disruption of the system. Therefore, the hash functions must produce unique outputs. Reducing the length of the outputs without losing this property will have a great impact on reducing the memory and transmission in the system.
4. Tracking problems can be prevented by randomizing the responses of a tag.
5. As mentioned in protocol 1,most authentication protocols are separated into two types:

   5.1 **Fixed access control:** tag replies with fixed messages.
   Advantage: simple and scalable design, low cost.
   Disadvantage: tracking problem due to the constant response of tags even if they are hashed. i.e. [89]

   5.2 **Randomized access control (RAC):** tag replies with varying pseudo-random messages. There are two types depending on the how the secret information is shared.

   5.2.1 Independent secret information (not shared).
   Advantage: if one tag is compromised the rest of the tags are not affected, solves tracking and cloning attacks.
   Disadvantage: un-scalable, if time-memory trade off used then memory requirements increase, requires extra computational workload and complex design.

   5.2.2 Share common secret information (all tags same secret).
   Advantage: scalable, solve tracking problem, less computational workload.
   Disadvantage: if one tag's secret is revealed to an attacker, then all the tags that share the secret are susceptible to cloning attacks. [90, 91]

3.6 Protecting RFID communications in supply chains

Protocol 6 [70].The authors present an RFID protocol for supply chains that emphasizes specific security requirements such as supply chain visibility. Although, supply chain management protocols are targeted towards specialized RFID applications, it is worth mentioning the major contribution points of the design. For example, protocols generally aim at single reader-tag communication whereas supply-chain protocols aim at multi-tag-multi-reader communication and security issues in supply chains have not yet been fully investigated. This is due to the processing nature of authenticating tags as bulks and not singularly. In addition, as the location and management of the shipment changes, different readers will in turn have to process the bulk independently. For instance, if there are three partners A, B and C and the shipment is currently managed by B, only A and B should be able to read the legitimate tags and not C or any other untrusted third party. The design of the protocol consists of two phases, the read and write schemes and protocols of such nature can be considered challenging. Below are the four security properties that RFID protocols must offer to supply chain management:

1. Unlinkability: is a weaker security property than untraceability since personal privacy is not considered in supply chain managements. The main purpose of unlinkability is to prevent an illegitimate reader from determining during interrogation if the tags are the same or different.
2. Authoritative access: a shipment consists of a bulk of tags. While that shipment is controlled by partner A, then only A should be able to reader the legitimate tags.
3. Authenticity of tags: for example, in a supply chain link, consisting of two partners A and B, B must only accept and read legitimate tags from A.

4. Visibility of a supply chain: should cover two main purposes, to allow only legitimate executives, e.g. managers, to track the process and during any phase to determine which partner is responsible for the processing of the tags at that current time.

The protocol's notation is as follows:

Notation	Explanation
C	tag's unique serial number
k (k_i, k_{i+1})	Access key – secret shared between Reader (Ri) and Tag
A	Tag pseudonym (a = C ⊕ k)
R	Secret mask – secret random number used for authentication
S	Status – if S = 1 then tag successfully processed else S = 0 pending or rejected.

Reader stores: k_1, k_i, k_{i+1}. Tag stores: *a*. Initially, the reader generates a random number N_R and sends it to the tag. The tag will then calculate *X* and forward it back to the reader that checks if *a* is valid i.e. $a = C \oplus k$ and try to locate a match for all the values in the database, received by the tag. If pseudonym and values are correct, R_1 and R_2 are then calculated, which correspond to the updating of the tag's pseudonym and forwarded back to the tag that will in turn verify the values and update accordingly. A note worth mentioning is that the protocol is divided into two phases. The read phase: Pi readers (partner in i-th position in the chain flow) must retrieve the information stored on the database for the corresponding tag once it has also successfully extracted the tag's serial number. Pi and the tag share the same access key. The Write phase: Pi must update the tag's access key so that the next partner, P_{i+1}, reader can securely interrogate the tag. Below is a diagram of the authentication process. (Original diagram [23])

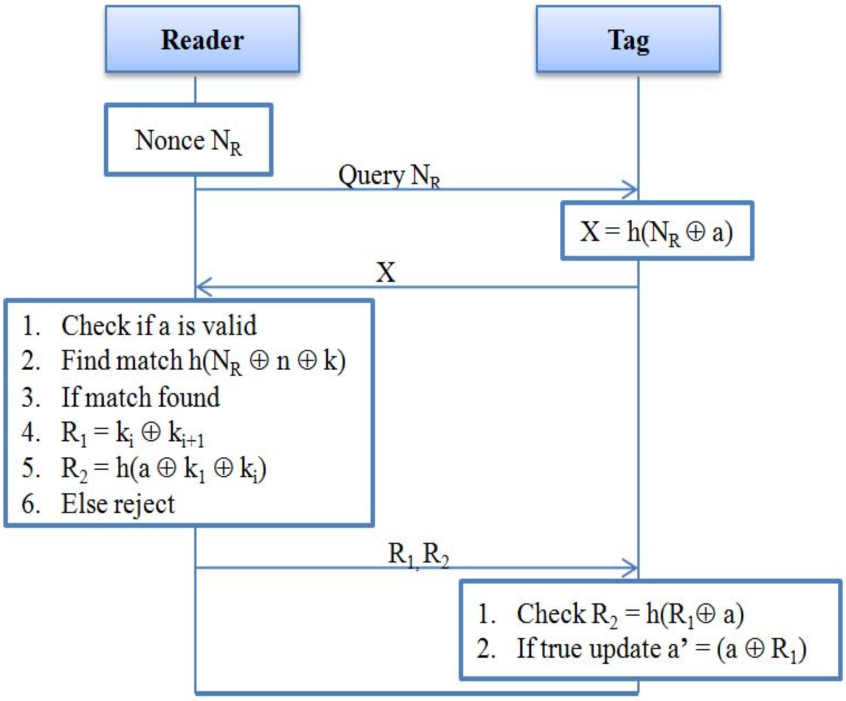

Reader
Tag
Nonce N_R
Query N_R
$X = h(N_R \oplus a)$
X
1. Check if a is valid
2. Find match $h(N_R \oplus n \oplus k)$
3. If match found
4. $R_1 = k_i \oplus k_{i+1}$
5. $R_2 = h(a \oplus k_1 \oplus k_i)$
6. Else reject
R_1, R_2
1. Check $R_2 = h(R_1 \oplus a)$
2. If true update $a' = (a \oplus R_1)$

Figure 11: Supply Chain Management Protocol Diagram

The authors prefer a simple design that it is efficient, practical and an approach that entails a novel advanced solution. Untraceability, which is the stronger notion of unlinkability, is disregarded and considered irrelevant in such applications. Untraceability is referred to as a personal privacy threat and unnecessary cost. Nonetheless, the authors neglect the fact that privacy does not only conform to issue concerning an individual but appears in other sectors such as organizational privacy[5] and industrial espionage [75]. In chapter 4, other weakness of the protocol will be exposed and analysed in further detail.

3.7 Authentication Pervasive Devices with Human Protocols

Protocol 7 [71].The theme of this paper is about identifying the limitations and similarities of human vs. pervasive devices and bearing those findings in mind, present a low-cost human-computer authentication protocol based on LPN (Learning Parity with Noise) problem. The authors illustrate a practical description of how EPCtags are similar to humans i.e. the limited amount of memory, both have low computational capacities, incompetent of recalling long passwords or processing complex calculations etc. The design of the protocol tries to compare the functioning environment and the capabilities of a passive tag with the theoretical scenarios of a user with access rights and without any additional help from external devices, logs onto an untrusted terminal and in the mean time is under observation. With this inspiration, Hopper and Blum presented a human-to-computer authentication protocol (HB), that forms the foundation of HB+. HB was proven to be unsecure against active adversaries but secure with passive i.e. eavesdropping. HB+ extends HB and the reduction of the LPN problem illustrates that the new protocol is now secure against both active and passive adversaries.

Brief description of the LPN problem: there exists a passive adversary exists that can observer several authentications and captures *n* rounds of the protocol. A matrix *A* exists that stores the challenges *a* generated by the reader and stores the responses by the tag as a set of vector *z*. The aim of the adversary is to compute another k-bit secret, ***x'***, such that the secret *x* and ***x'*** are similar or identical and finally yield a set of similar responses as *z*. The LPN problem is regard as NP-Hard, referred to as the Minimum Disagreement Problem and shown by Hober and Blum to be log-uniform and pseudo-randomizable [92]. The goal of the adversary is similar to the LPN problem [6]. According to the authors, HB is an identification scheme that preserves privacy since the tags identities are secretly preserved since an eavesdroppers can only capture an instance of the LPN problem and not the ID itself.

Below is a brief description of the HB protocol and the improved version, the HB+ protocol. In the HB version, both reader and tag store a k-bit secret *x*. In the HB+ version, the tag and reader store *x* and an additional k-bit secret *y*. The number of rounds required for authentication is the also the security parameter. Step 1, the reader generates a random challenge *a* and sends it to the tag while the tag generates a noise bit *v* (*v* is equal to 1 with a probability h $\leftarrow$ [0,1/2]). Step 2, the tag calculates $z = (a \cdot x) \oplus v$, and sends the result *z* to the reader. Step 3, if the parity bit of the tag is correct by calculating $a \cdot x$, then the reader accepts otherwise rejects. After a number of rounds, the reader authenticates a tag and accepts if the calculation yields the same result as z. In general, the HB protocol is simple and consists of only XOR, and AND operations. According to the authors, *v* which is the noise parameter can be produced by cheap/physical methods such as shot noise, oscillation jitter etc. (Original diagram and description [80]).

[5] Organizations, governments and industries frequently enforce necessary means in order to prevent disclosure of secret information such as trade secrets, activities and movement of goods, products etc.

[6] LPN Problem [71]: find x' such that $|(A \cdot x') \oplus z| <= hq$. Notation: x = random k-bit vector, q = number of rounds, A = matrix that stores challenges and responses (a and z), h = a noise parameter.

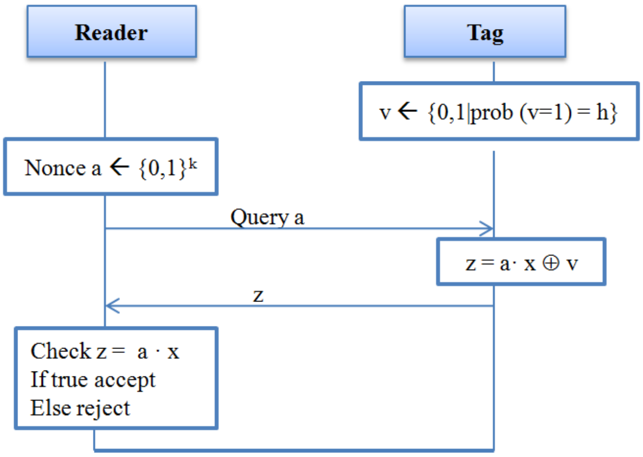


Figure 12: HB Protocol Diagram

The HB+ protocol is very similar with HB. The main differences are:

1. The tag generates a random "blinding" vector *b* at each query.
2. The symmetric key consists of an additional shared k-bit vector *y* stored in both the tag and reader.
3. The tag now computes z = (a · x) ⊕ (b · y) ⊕ v.

The diagram below shows a single round of the protocol. The authors mention that the protocol's format is similar to the Fiat-Shamir identification, i.e. commit, challenge, response. HB+ does not require much additional computation or memory as *a* and *b* are partially stored and the only extra storage is required for *y*. (Original diagram [71]).

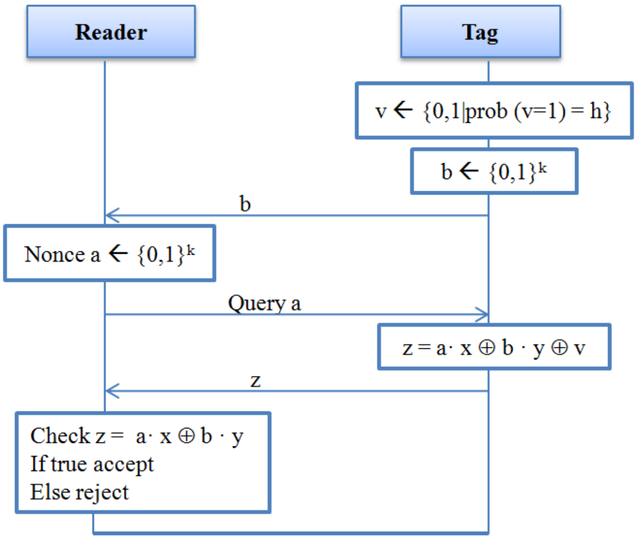


Figure 13: HB+ Protocol Diagram

The security description of HB+ is as follows. With the additional "blinding" vector *b*, the adversary cannot extract any information on *x* or *y* even if the adversary can observe *b* and *(b · y) ⊕ v* and protects against active attacks that allow *x* to be leaked (*x* is independent of *y*). In addition, both passive and active adversaries are incapable of obtaining *a · x* due to the *(b · y) ⊕ v*. The aim of the security proofs are to illustrate that if an adversary can extract the secret *y*, then the attacker can also solve the LPN problem.

The authors themselves pose questions concerning the security of the protocol when considering a two-round version of HB+ in a duplex channel where *a* and *b* are sent simultaneously. If, as stated and proven in the paper, the security of the HB+ protocol is based on the LPN problem but the hardness over random instances remains an open question, how will the overall security and efficiency of the protocol be affected. The HB and HB+ protocols are lightweight designs that could be implemented on various software and hardware platforms. However, the security properties have been proven weak.

3.8 A Scalable and Untraceable Authentication Protocol for RFID

Protocol 8 [72].The authors claim that prior to the proposal of this protocol, it was impossible to achieve untraceability and scalability. As seen in other protocol descriptions, these two properties are possibly the hardest to achieve and often designs require a trade-off between them. This protocol offers other security properties such as replay attacks, cloning attacks spoofing etc, supports ownership transfer and can be implemented in a multi-tag-reader environment.

Below is a list of paper's main points:

1. Most of the computational workload should be handled by the back-end of the system for both large and small systems.
2. Pseudo-EPC are implemented into the tags memory and not the actual EPC code (tag ID). An adversary able to extracts the EPC code, can also classify the tagged products i.e. value, merchants etc. However, protocol designs considering this method must ensure that the back-end servers are able to link the pseudo-EPC with the actual code of the tag, promptly and efficiently.
3. Tags that respond with static messages including information about their ID have a higher chance of traceability. Otherwise, if the responses do not include information about a tag's ID and change for every session i.e. dynamic, then traceability is solved but un-scalability is then the main issue.
4. Approximately four hash functions are used and a common shared secret between tag and reader. The secret authenticates the reader to the tag and vice versa. The reader can authenticate only tags containing the secret. The authors claim that this is a 'novel' approach in RFID protocol designs.
5. Timestamps are used to prevent replay attacks.
6. An extra property is introduced, item privacy. An active adversary is unable to determine the tagged product's content or value. Other protocols include this property in other privacy properties such as user data confidentiality.
7. The authors argue that the protocol provides: synchronization, ownership transfer, untraceability, forward secrecy, untraceability during a valid session, untraceability, scalability, prevents spoofing and item privacy.

The brief description of the protocol's operation is as follows. The tag stores the shared secret *k*, pseudo-EPC (*ID*), access password *PSW*, and last timestamp sent by reader, *TS*. The reader stores the shared secret key *k*, and the ID and PSW for each tag. The reader generates a timestamp and computes the hash of *k* and the timestamp i.e. *X = h(k,TS)*, and sends *X*, *TS* to the tag. The tag then compares *TS* sent from the reader with *TS***'** and if *TS*>*TS***'** is true then it computes *X' = h(k,TS)* and compares once again *X* with *X***'** in order to authenticate the reader. If *X*=*X***'** and *TS*>*TS***'** the tag sends the hashed ID which the reader then forwards to the database including a new timestamp *TS*. The database then computes *X = h(ID,PSW)* and sends *X, TS* to the reader which forwards the values to the tag. In the mean time, the database updates the new *ID***'** *= h(ID,PSW,TS).* The tag once again compares *TS* and computes *X***''** *= h(ID,PSW)* and compares X (received) with X'' (computed). If the values are equal, the tag then also updates the new *ID***'** *= h(ID,PSW,TS)*. The final step is to update TS**'** (the last generated timestamp). (Original diagram [72]).

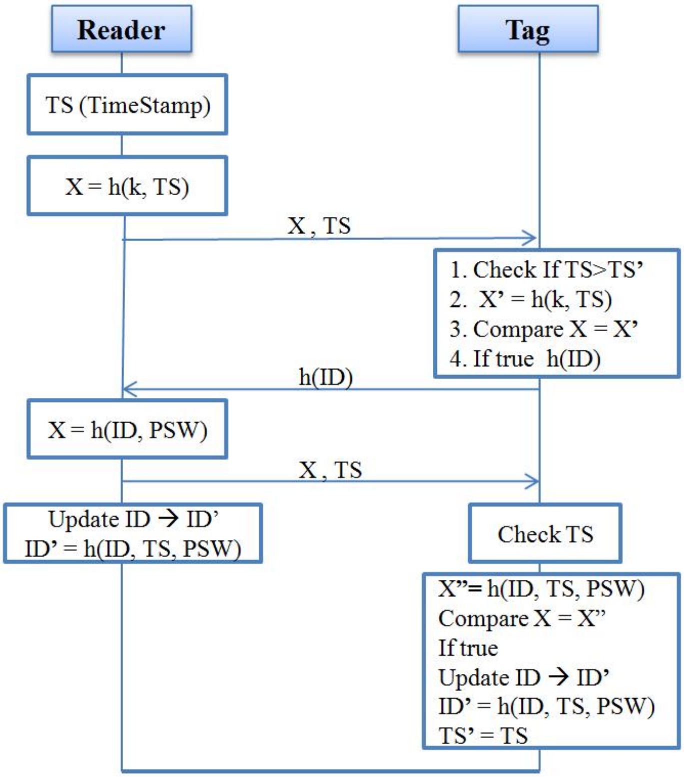


Figure 14: Protocol Diagram that uses Timestamps and Passwords

One of the steps in the procedure that is different to most protocols is the authentication of the reader that occurs twice, at the beginning and the end of the protocol's execution. The robustness of the protocol's design and the security properties are vaguely examined and conclusions are drawn without investigating significant details in both areas. In chapter 4, a number of unpreserved privacy and security properties are shown.

3.9 RFID Authentication Protocol with Strong Resistance against Traceability and Denial of Service Attacks

Protocol 9 [73].The authors propose an authentication protocol claiming to be secure against DoS attacks, spoofing attacks and location tracing. The main RFID security problems are separated into two categories, traceability and information leakage. However, the authors do not mention that traceability is a security attack whereas information leakage is a privacy issue. Location tracing is also a highlighted property and according to the description, it is similar to breadcrumb threat mentioned in chapter 2. A security threat model identifies the general weaknesses in an RFID system and the authors then analyses possible vulnerabilities in RFID authentication protocols. Kang et al [73], divided the RFID communication protocol into three layers, the application, communication and physical layer. The authors stress that all three layers suffer from location tracking attacks and they propose an authentication protocol aiming at the security of the application layer.

There are two versions of the authentication protocol, the basic and alternative protocol. The alternative differs in the order the authentication process takes place. The reader authenticates the tag in the basic protocol whereas in the alternative, the tag authenticates the reader. The alternative performs two hash computations and increases the number of communication messages in comparison to the basic version that requires four hashes and less messages. Below is the description of the basic protocol design. (Original diagram [73])

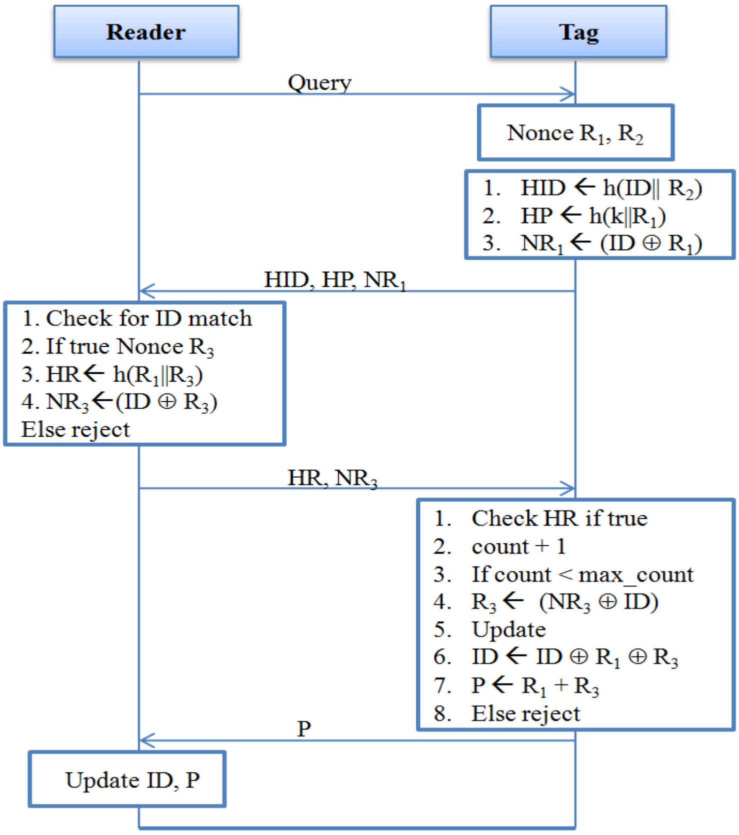


Figure 15: Protocol Diagram that claims strong resistance against DoS

The reader and the tag stores the secret key *k,* and the *ID*. The reader sends a request query to the tag. The tag then checks the status of the session, if it is true (session already in progress) then it sends back the same randomly generated nonce R_1 and R_2. If the status is false, the tag generates R_1 and R_2 at random and sets the status to true. Next it computes three messages, *HID, HP* and NR_1. The three messages are then forwarded to the database via the reader, that will check for a valid match for the *ID* and *k*. If no valid ID is found, the session ends. Otherwise, the server computes a random nonce R_3 and calculates $HR = h(R_1||R_3)$ and NR_3 and forwards the values to the tag. The tag has a counter that increases as each message is received. The counter when it reaches the maximum value, it terminates the session and waits for a new one to begin. Next, if the counter is less than the maximum, the tag checks if *HR* is correct. If the values match, the tag sends a confirmation value *P* that the values are correct. The final step consists of the server updating the *ID* and *k* if the message sent by the tag is correct. The tag also changes the *ID* and *k* and sets the status to false, i.e. session complete. Otherwise, it halts the procedure and identifies it as an attack.

Practical points and new terminology gathered from the paper:

1. Pre-emptive locking: in situations where a tag receives a query while it is already in an authentication process, it can either ignore or start a new authentication session with the sender of the query. Both actions carry consequences, the former can result in the interaction with an attacker, preventing by constant denial responses and ignoring the state of the tag, from interacting with a legitimate reader. Instead, if the tag accepts every message, it can lead to a DoS attack. The authors recommend a solution to this type of attack (point 3).
2. Stealth bombing: similar to DoS attack. The attacker sends tampered messages during authentication allowing the tag to constantly generate new random nonces and initialize new session, resulting in the continuous state of an ongoing authentication process.
3. If the randomly generated nonces, i.e. R_1, R_2 are kept identical without changing during an ongoing session, it prevents an attack from proceeding when requests are sent while in

process of authentication. Stealth bombing and pre-emptive attacks are supposed to be prevented with this technique.

4. Threshold counter can be used to increase the counter of received message and when reaching the maximum value, the session halts and waits for a request to re-start a new session.
5. Timestamps are also used in this protocol as an alternative to timers that are considered expensive for low-cost RFID tags.

3.10 Dynamic Key-Updating: Privacy Preserving Authentication for RFID Systems

Protocol 10 [74].Lu et al. in their paper present the SPA (Strong Privacy Authentication) protocol. The protocol's design is according to the tree-based approach including a novel key update procedure. SPA claims to be secure against passive, active, and compromising attacks. This paper provides a clear description of how the tree-based approach works and introduces an attack model by Avoine [77] to prove the protocol's security.

According to the authors, tree-based protocols are supposed to be the most practical, on the contrary they do pose a serious issue when it comes to the usage of keys during tag and reader authentication. Generally, in such protocols there are a number of keys corresponding to one tag and each one of these keys are stored in turn and individually on a single node. The path of keys from the root up to the tag, is the set of keys designated to the corresponding tag ending at the last leaf. The key storage infrastructure of such protocols, introduce weaknesses in the design, which is static, and does not update keys allowing tags to share common keys. If an attacker tampers with one tag that shares common keys with another tag, it will affect the linked tags as well. SPA proposes a dynamic key update mechanism that solves this problem. Below is the description of the design.

SPA protocol design consists of four components (Original diagram [74]):

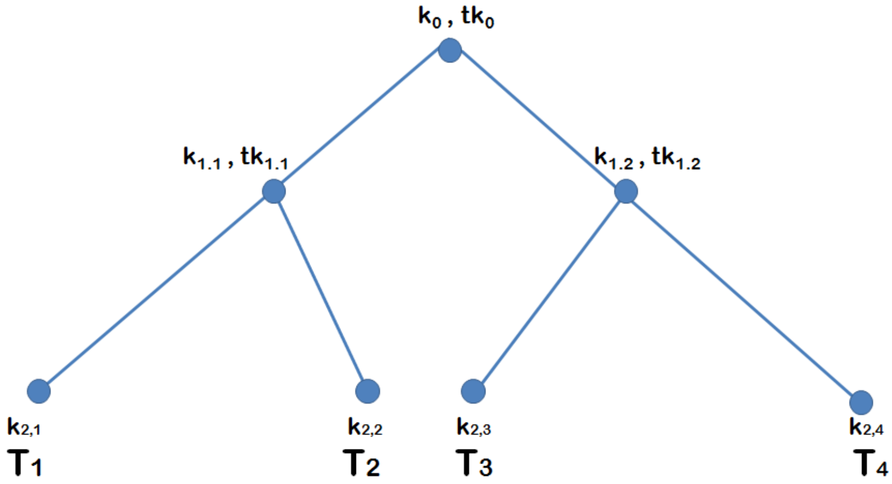


Figure 16: Distribution of keys for a tree-based protocol

1. System initialization: The reader (including the back-end server) assigns each tag to a unique leaf node in the balanced binary tree. There are two sets of keys, the current key (k) and the temporary key (tk), placed on each 'non-leaf node'. It is the reader responsibility to generate the random keys and at the initialization phase the current key is equal to the temporary key.

2. Tag identification: the authentication of the reader and tag includes three rounds. First, the reader sends the tag a query request and a random number. Second, the tag generates another random number x_2, and computes the hash function of each key (k^{i}_{0}... k_n) denoted to that specific tag with the random number x_1 sent by the reader. The final output is F ($F = x_2, h(k_i , x_1)$). The reader then starts to

authenticate the tag upon receiving F. An algorithm is used to traverse from the root up to the leaf and compares the output of each calculation with the keys at each node. According to the results of the calculation, the reader will form a path that ends with identifying the tag.

3. Key-updating: Once step 2 is complete then the key-update procedure begins. A new key is generated from the hash value of the old key, i.e. k = h(k') and the temporary key, tk, is used to store the old key. This mechanism prevents the interruption of other ongoing authentication procedures. Once the updates are over, two messages are sent to the tag, one is used as a tag-to-reader authentication, and the second is a synchronization messages that includes the status of key updates.

4. System maintenance: This component ensures successful tag addition and removal from the system, e.g. enrolment and withdraws methods. When a new tag is added unto the tree, the reader first locates an empty leaf node and assigns the tag to that leaf. The keys allocated from the root up to the new leaf, are the tag's corresponding keys. When tags want to be removed, the reader vacates the associated leaf and sets the parent's node status to 1, identifying that the leaf node is empty.

The authors mention several security requirements for RFID systems: Privacy, forward secrecy, untraceability, cloning and compromising resistance. Two properties are presumed as essential in order to prevent active attacks:

1. Correlated-exposing: accessing previous messages send by the compromised tag. Similar to backwards traceability.
2. Past-exposing probability: the probability of tracing other tags according to the gained information of the compromised tag. Similar to cloning/spoofing attacks.

Avoine's attack model [77], briefly is a cryptographic game of an adversary trying to distinguish between two tags, i.e. T_0 or T_1 with probability greater than 50%. The adversary has access to four oracles, Query, Send, Executive, and Reveal. If the adversary wins the game, then the protocol is stated as insecure. However, the authors prove with the above model, that their protocol is secure against compromising attacks and guarantee several security properties. The computation time required in this protocol is O(logN) (N = number of tags) tree walking steps.

3.11 A Lightweight RFID Protocol to protect against Traceability and Cloning attacks

Protocol 11 [75].Dimitriou's proposes a lightweight authentication protocol worth considering since it is based upon a simple design and techniques, suitable for low-cost RFID tags despite the failure to achieve several security properties. The author first analyses a simple protocol that claims to be secure against a number of possible attacks except from replay attacks which are the easiest to prevent. An enhanced version is then presented in order to fill in the gaps of the simple design and provide stronger security. Both versions are based on a challenge-response technique. The significance of mutual authentication in a protocol is highlighted and so the design includes a tag-to-reader and reader-to-tag authentication process. This presumes the protection against cloning and tracking attacks. For example, tag-to-reader prevents cloning attacks and reader-to-tag authentication prevents unauthorised queries by malicious readers.

The diagram below demonstrates the enhanced version of the protocol. Both versions apply cryptographic hash functions for indistinguishability of messages and to construct the new ID. The tag stores the ID and the reader stores the secret key and the IDs for all the tags in the system. The extended protocol applies a tag-to-reader message to prevent eavesdropping. The reader sends a request to the tag and a randomly generated nonce X_R. The tag the replies with its hashed *ID*, a new nonce X_T and the hash of the two nonces, X_R and X_T with the tag's *ID* as a key. The reader will transfer these values to the database that will check if the values are correct, locate a matching *ID* and update the new *ID*, i.e. including the hashed value of the two nonces under the new key (ID_{i+1}), $D = h_{ID+1}$ (X_R

+ X_T). The reader then forwards the value D to the tag which computes $D' = h_{ID+1} (X_R + X_T)$ and compares with the received D. If the values match the tag updates the new ID with D', otherwise if no match is found, it rejects the final message. (Original diagram [75]).

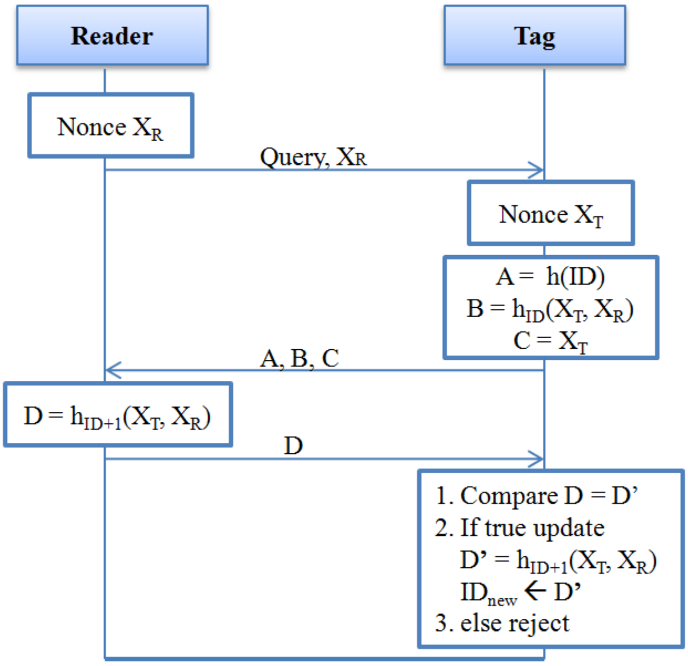


Figure 17: Dimitrious Lightweight Protocol Diagram

Other interesting points the author mentions, is when designing mutual authentication protocols, reader-to-tag authentications should avoid readers that can only authenticate themselves to tags with secret information such as the tags ID or secret key k. However, designs are usually based on a tag-to-reader authentication which is a simpler process and if mutual authentication is applied in a protocol, the limitation has been solved with various techniques. For example, a tag includes a random number generated and sent by the reader in the message that contains the tag's ID. Moreover, the enhanced version mentions that the third message is used to prevent desynchronization. In chapter 4, several possible attacks and security weakness of the enhanced version are demonstrated.

3.12 Minimalistic Cryptography for Low-Cost RFID Tags

Protocol 12 [76].The protocol presented by Juels is a one-time-pad security scheme based on the use of pseudonym updates. The aim of the author is to prove that implementing cryptographic functionalities in RFID systems, that form a great challenge to researchers, is not compulsory for achieving the required security in RFID systems. The author claims that the presented protocol captures real-life situations and is suitable for low-cost tags. The assumption is based on adversaries that have limited powers and interfere with the communication channel periodically. For this reason, the security model described in this paper is supposedly reflecting real-life issues in RFID applications, were adversaries have less computational powers and limited access to random oracles, in contrary, with traditional cryptographic schemes. Thus, the protocol eliminates standard cryptographic functionalities and hash functions. Another aim of the design is to rely on the memory of the tag instead of computational resources, whereas up to now protocols have been designed considering both factors.

The protocol works as follows. Each tag contains a small number of pseudonyms that corresponds to the tag's IDs. During an execution of the protocol, the tag rotates through these pseudonyms, and releases a different one on each reader query. The readers will then authenticate the tags since they store the complete pseudonym set for all the tags in the system. In more detail, the pseudonyms operate as keys and are the main point in this scheme. The tag sends a pseudonym *X* to the reader and the reader in turn replies with another pseudonym *Y*. There is an association between *X* and *Y* since *Y* is the only possible response to a given *X*. The reader must first authenticate itself to the tag and only then, the tag authenticates itself back to the reader by generating an extra pseudonym key, *Z* (mutual authentication). The pseudonym *Z* has the same restrictions as *Y*. Once the authentication process has successfully completed, the reader updates all three values, *X, Y, Z*. This prevents the reduction of the tag's integrity by preventing tracking attacks evolving from communication channel observation, over a long period. Although, an adversary can eavesdrop while updates are in process and can tamper with the system in general, Juels uses one-time pad technique to prevent this by restricting access to any of the three updated values. One-time pads do not use encryption but work in a similar way by screening the updates with pads that can resemble keys.

There design of this protocol is exemplifies interesting observations including both strong and weak points. As seen with other protocols one of the main goals is scalability, in this protocol the author states that the communication cost is linear, meaning as tags increase, scalability decreases and cost increases. Pseudo-throttling is a practical method that does not require much computation but relies heavily on memory and efficient updating of the pseudonyms is essential. As mentioned above, the aim for the design is to reflect real-life security scenarios and thus the protocol is later on proven weak even for low-cost RFID systems.

4. Security analysis and comparison of the protocols

This chapter focuses on the security analysis of the twelve protocol described in chapter 3. All the above protocols have been proven vulnerable to a number of attacks, provide weak privacy, or perform operations that require memory and computation resources that low-cost tags cannot afford. However, not all RFID applications require the same level of security and thus, a variety of proposed protocols are analyzed. Nevertheless, the main point of this section is to understand, under what circumstances can a protocol offer or limit a security property and compare in terms of low-cost RFID tags for general applications. Other papers providing similar comparisons are frequently referred to with the intension of enhancing the analysis of this chapter. The final outcome includes an overall evaluation of the protocols security and privacy achievements, in order to extract essential points and avoid security pitfalls that will enhance the design of the proposed protocol in chapter 6.

An attacker able to successfully perform a security attack will also form a corresponding privacy issue. Below is a table of the main security goals an RFID protocol must achieve [7]. Other privacy and security properties are: tag anonymity, man-in-the-middle attack resistance, user data confidentiality, forward secrecy, cloning prevention, untraceability, replay attack prevention, and compromising resistance.

Goals	Description
Prevent Denial of Service (DoS)	In situations where DoS attacks cannot be prevented other mechanism that can cope with the attack must be available with the purpose of sustaining the availability of the system. For example, DoS prevention is a crucial property for medical applications such as patient monitoring [7].
Prevent tracking and identification	This goal comes in two flavours. First, when humans are involved in the system, it is extremely important that both tracking and identification is prevented at all times. Secondly, if the system involves industrial purposes e.g. pallet tracking, it can be disregarded depending on the desired security level of the system [7].
Prevent illegitimate access	The level of data integrity is linked with the possibility of illegitimate access. If an adversary can access the stored data and change it, then the data on the system will no longer be accurate or controlled. This is an important property to attain for all RFID applications [7].
Prevent tag imitations and 'valid' duplicates	Authenticity is the main key to prevent tag imitations. Systems must ensure that tags emitting information linked with any type of identification cannot be compromised or duplicated by an adversary [7].
Preserve data security	Data held in an RFID system must be kept confidential by preventing illegitimate reading of such information. A situation of data leakage will accentuate privacy implications and issues [7].

Table 3: Desired Goals for RFID Protocols

4.1 Protocol 1

The first version of EC-RAC claimed to be scalable, anti-cloning and provide tag anonymity. The protocol uses a randomized access control design and tags reply with a pseudorandom message that changes at each session. If the secret is specific to each tag and not common, scalability of the system is possibly compromised since tags will have to frequently update their secrets. However, tracking issues are supposedly solved. Other approach such as time-memory trade off resolves scalability issues but requires a large amount of memory. If common secrets are shared, then the tracking problem is still prevented but compromising one tag will compromise other tags sharing the secret and exposes them to associated attacks such as cloning, spoofing etc [69]. Recalling the protocol notation from chapter 3, below is EC-RAC version 1 attack description [23]:

Secret keys = x_1 (ID), x_2 (Password)
Public keys = x_1P (ID Verifier), x_2P (Password verifier)
Other publicly know points on the elliptic curve → y (only the reader knows), $Y = yP$, P.
Calculations used during the session → $T_1 = r_1P$, $T_2 = (r_1 + x_1)Y$, $u = r_1x_1 + r_2x_2$
In order to mount the attack the following steps need to be carried out in sequence.

1. The adversary has to complete two protocol rounds with the tag (the values in the second round are primed).
2. In both sessions, the same random number must be used i.e. r_2
3. The attacker has to calculate $u – u' = x_1 \cdot (r_1 – r_1')$ and $T_1 – T_1' = P \cdot (r_1 – r_1')$
4. Calculate the inverse v – v' modulo
5. Obtain $x_1^{-1}P$ → tag is traceable

Due to the above mountable attack, an adversary can identify a tag thus eliminating the property of untraceability. Recalling from the protocol description, although x_1P is the public key and the ID verifier, the authors store the value securely in order to prevent attackers from obtaining it. As shown from the steps above, an attacker is able with two runs of the protocol, to distinguish a tag and for this reason; tags can be tracked [23].

The protocol achieves forwards secrecy and prevents replay attacks because the tag generates a random nonce r_1, which changes for every session and is used for the messages sent to the reader. DoS attacks are also prevented given that an attacker blocking the last message will not result in desynchronization since the reader and the tag do not perform any updates during the last messages of the authentication process [23].

4.2 Protocol 2

The second version of EC-RAC introduces a group of six protocols all containing four components in different order. The table below illustrates the claimed security properties, scalability and type of authentication of the six protocols. (One-way: tags only require authentication, mutual: both tag and reader authenticate themselves). (Original table [66]).

Components	Protocol 1	Protocol 2	Protocol 3	Protocol 4	Protocol 5	Protocol 6
Scalability Untraceability Authentication	Small ✓ One-way	Large X One-way	Large ✓ One-way	Small ✓ Mutual	Large X Mutual	Large ✓ Mutual

Table 4: Claimed properties for the six authentication protocols

Deursen et al [81], contradicts the statement of the authors that claim, if one scheme satisfies a security requirement, when combined with others it will remain secure. The protocols are shown to be vulnerable to man-in-the-middle attacks and do not satisfy untraceability or authentication [81]. The authors of the EC-RAC II, draw conclusions disregarding several important points that allow the protocols robustness to be questionable. For example, protocols 4, 5, 6 claim to be security due to the server's authentication that uses generated random numbers. Another vague statement concerns an attacker claimed to be incapable of duplicating tags even under the condition were a vulnerable server reveals the stored information. The authors justify this by storing the secret information on the tags and not on the server. Recalling, that tags have limited amount of memory and servers are able to inherit strong security techniques, this might not be a practical solution for RFID protocols [81, 88].

Protocols 4, 5, 6: these protocols apply a challenge-response technique to authenticate the server with the tag. This technique can be used by an attacker as an oracle to obtain critical information and eventually impersonate a tag. The components are not independent and they use the same server's secret key y (scalar point only known to the server) for their own purposes i.e. tag-to-server authentication and server-to-tag authentication consequently, and according to [81] the security is questionable. All six protocols are vulnerable to a man in the middle attack on their ID transfer scheme and the tags are traceable. An adversary performs a man-in-the-middle attack on two communication sessions and obtains $r_{t1}P$, r_{s1} and $(r_{t1} + r_{s1}x_1)Y$. He then computes $r'_{t1}P + r_{t1}P$ and subtract r_{s1} - r'_{s1}, and adds the two different responses. The results are then sent to the server which will accept if $x_1 = x_1$' else reject. If the server accepts, then the tag is the same as the one in the first attack [81]. (Original diagram [81]).

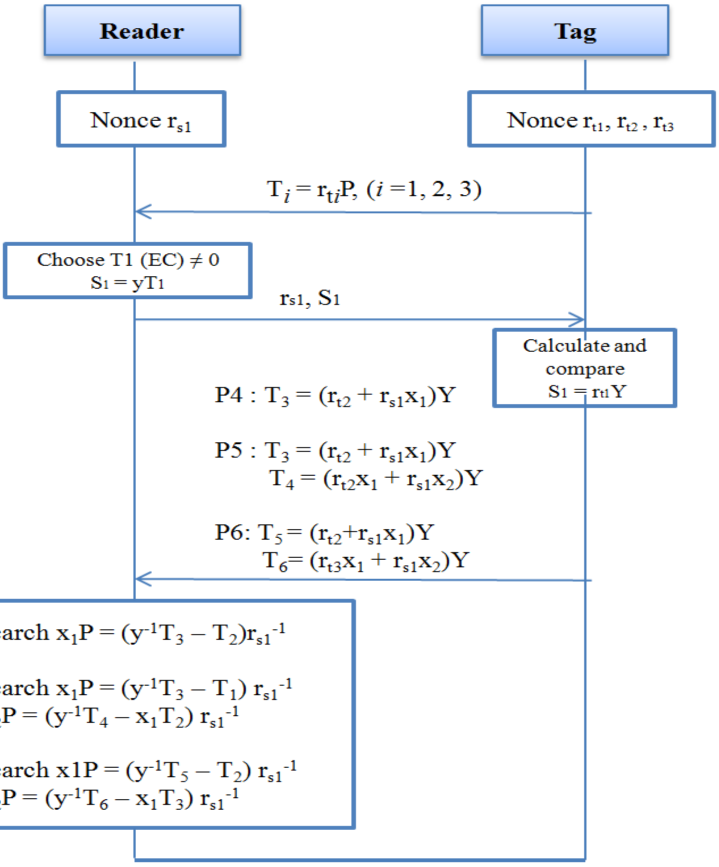

Reader
Tag
Nonce r_{s1}
Nonce r_{t1}, r_{t2} , r_{t3}
$T_i = r_{ti}P$, $(i = 1, 2, 3)$
Choose T1 (EC) ≠ 0
$S_1 = yT_1$
r_{s1}, S_1
Calculate and compare
$S_1 = r_{t1}Y$
P4 : $T_3 = (r_{t2} + r_{s1}x_1)Y$
P5 : $T_3 = (r_{t2} + r_{s1}x_1)Y$
$T_4 = (r_{t2}x_1 + r_{s1}x_2)Y$
P6: $T_5 = (r_{t2} + r_{s1}x_1)Y$
$T_6 = (r_{t3}x_1 + r_{s1}x_2)Y$
P4: Search $x_1P = (y^{-1}T_3 - T_2)r_{s1}^{-1}$
P5: Search $x_1P = (y^{-1}T_3 - T_1)\, r_{s1}^{-1}$
$x_2P = (y^{-1}T_4 - x_1T_2)\, r_{s1}^{-1}$
P6: Search $x1P = (y^{-1}T_5 - T_2)\, r_{s1}^{-1}$
$x_2P = (y^{-1}T_6 - x_1T_3)\, r_{s1}^{-1}$

Table 5: Attacks on Protocols 4, 5, 6

Protocols	Random integers	Send to server	Challenge from server	Compute
Protocol 4	r_{t1}	$r_{t1}P$	r_{s1}	$T_3 = (r_{s1}x_1 + r_{t2})Y$
Protocol 5	r_{t1},r_{t2}	T_1, T_2	r_{s1}	$T_3 = (r_{s1}x_1 + r_{t2})Y$ $T_4 = (r_{t2}x_1 + r_{s1}x_2)Y$
Protocol 6	r_{t1},r_{t2},r_{t3}	T_1,T_2,T_3	r_{s1}	$T_4 = (r_{s1}x_1 + r_{t2})Y$ $T_5 = (x_1r_{t3} + r_{s1}x_2)Y$

Protocols 4, 5, 6 do not satisfy untraceability. An attacker has to eavesdrop on the communication line two times in order to obtain significant information. 1st time = ($r_{t2}P$, r_{s1}, ($r_{t2} + r_{s1}x_1$)Y), 2nd time = ($r'_{t2}P$, r'_{s1}, ($r'_{t2} + r'_{s1}x'_1$)Y). The attacker then computes and compares the results with the aim of identifying the tag he eavesdropped on first. An oracle is used in the process, were the attacker sends ($r_{s1}r'_{t2} - r'_{s1}r_{t2}$)P and receives ($r_{s1}r'_{t2} - r'_{s1}r_{t2}$)Y. Thus, the attacker can accomplish his goal in distinguish the tag. The protocols also fail to preserve forward or backwards untraceability properties. The protocols also fail to accomplish authentication between the tag and the server since an adversary can obtain x_1Y, x_2Y from computing the public keys x_1P, x_2P. In all the last three protocols (4, 5, 6) the adversary has to choose random integers and send information back to the server in order to receive a challenge back and then carry on with further calculation. Finally, the attacker will have all the necessary information to break the authenticity of the system [81]. (Original diagram [81]).

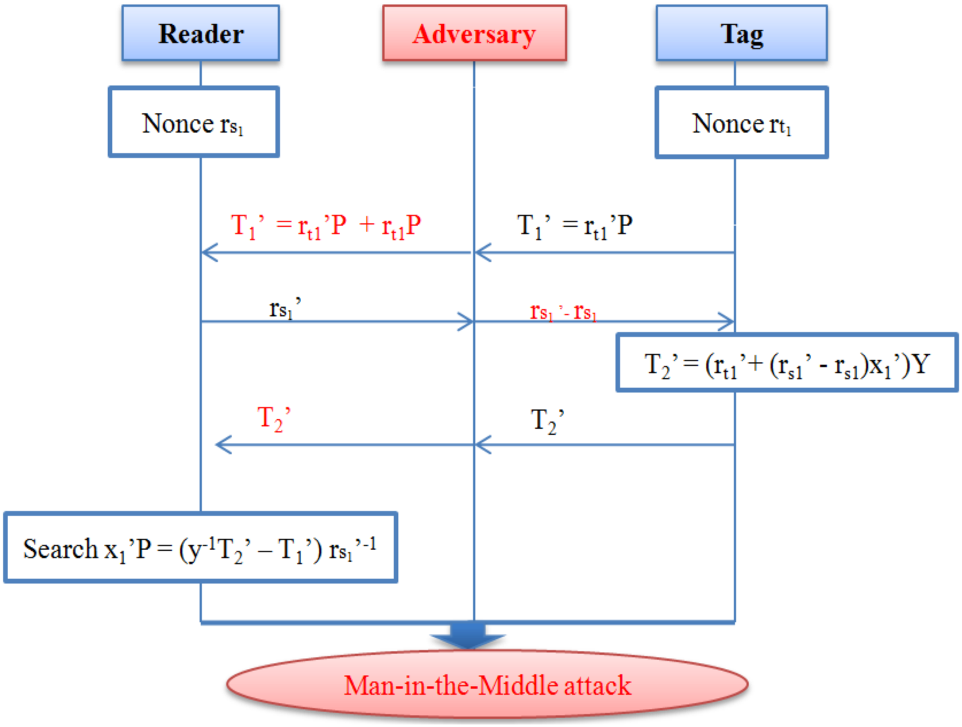

Reader
Adversary
Tag
Nonce rs_1
Nonce rt_1
$T_1' = r_{t1}'P + r_{t1}P$
$T_1' = r_{t1}'P$
rs_1'
$rs_1' - rs_1$
$T_2' = (r_{t1}' + (r_{s1}' - r_{s1})x_1')Y$
T_2'
T_2'
Search $x_1'P = (y^{-1}T_2' - T_1')\, rs_1'^{-1}$
Man-in-the-Middle attack

Table 6: Man-in-the-middle attack

4.3 Protocol 3

The final version of EC-RAC III claims to improve the previous versions by eliminating the use of the same keys for the ID-Transfer and Pwd-Transfer schemes. The most noticeable change is the utilization of non-linearity [79]. The authors state that the protocols are wide-strong privacy preserving. Meaning, that the design prevents wide attackers able to obtain the results of the server's verification, i.e. accept or reject and strong attackers that are able to extract secret information from tags allowing the attacker to constantly track the specific tag.

This version was proven by Fan et. al[79], to be at most narrow-strong privacy preserving. Generally the main mistake of the protocol is the use of the random challenge r_s. This value does not mask the secret keys x_1 and x_2 and thus attackers will eventually succeed in identifying a tag. The authors justify their findings with the following three types of attacks [79].

Attack on protocol 1: in the presence of a strong attacker able of accessing x_1 = tag's ID has the ability to mount a man-in-the-middle attack and track a specific tag by sending and receiving appropriate information. The attack is as follows; an adversary performs a man-in-the-middle attack and replaces the value r_s with r_s' and T2 with T2', then accordingly the reader accepts if $x_1 = x_1$'. If the process is accepted then the adversary determines if the tag is the manipulated one [79]. Below is the diagram that illustrates the man-in-the-middle attack for protocol 1. (Original Diagram [79]).

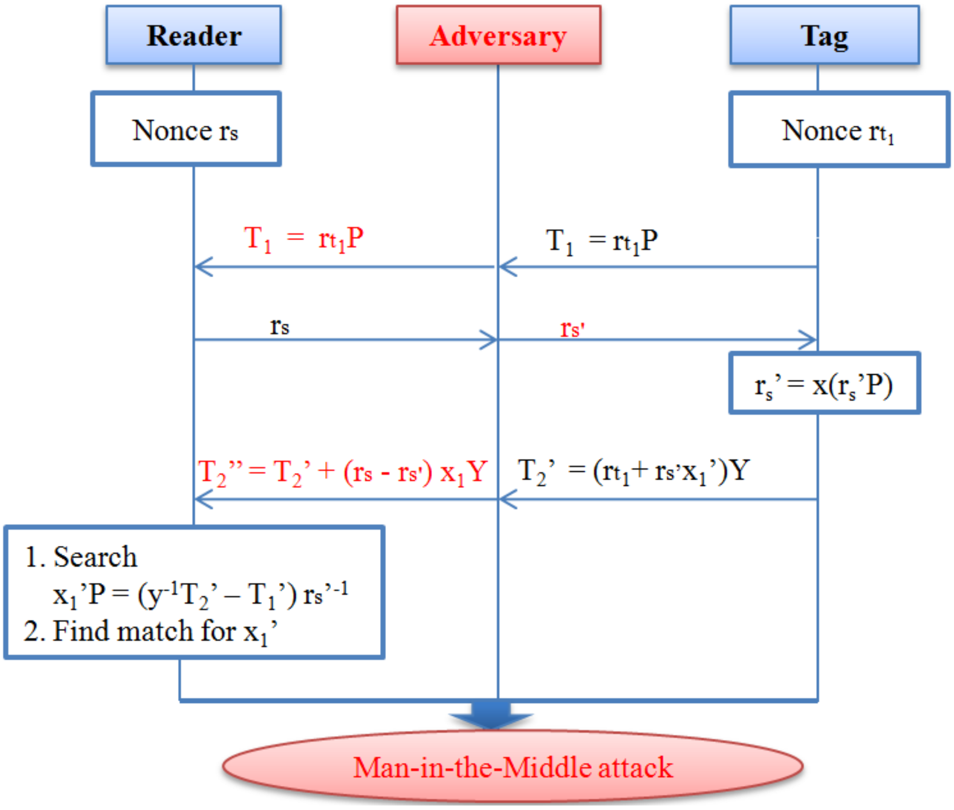


Table 7: Man-in-the-Middle attack - Protocol 1

Attack on protocol 2 and 3: the server randomly generates a value r_s at the beginning of the protocol run. This value can be manipulated by an adversary because the system never checks if it generates a multiple of the order P and if $r_s = 0$. If the latter case stands, then an adversary can impersonate a server in order to obtain the relevant information to finally, impersonate a legitimate tag. The adversary already knows T_1, T_2 so he forwards the changed r_s and receives T_3 and T_4 from the server. With T_1, T_2, T_3, T_4, the adversary can now perform a man-in-the-middle attack on the second run and determine if the tag observed in the first protocol run is the same as the second. The server accepts if $x_1 = x_1$' meaning, the tag from the first run is the same as the tag from the second run [79].Below is the diagram that illustrates the man-in-the-middle attack for protocol 2 and 3. (Original diagram [79]).

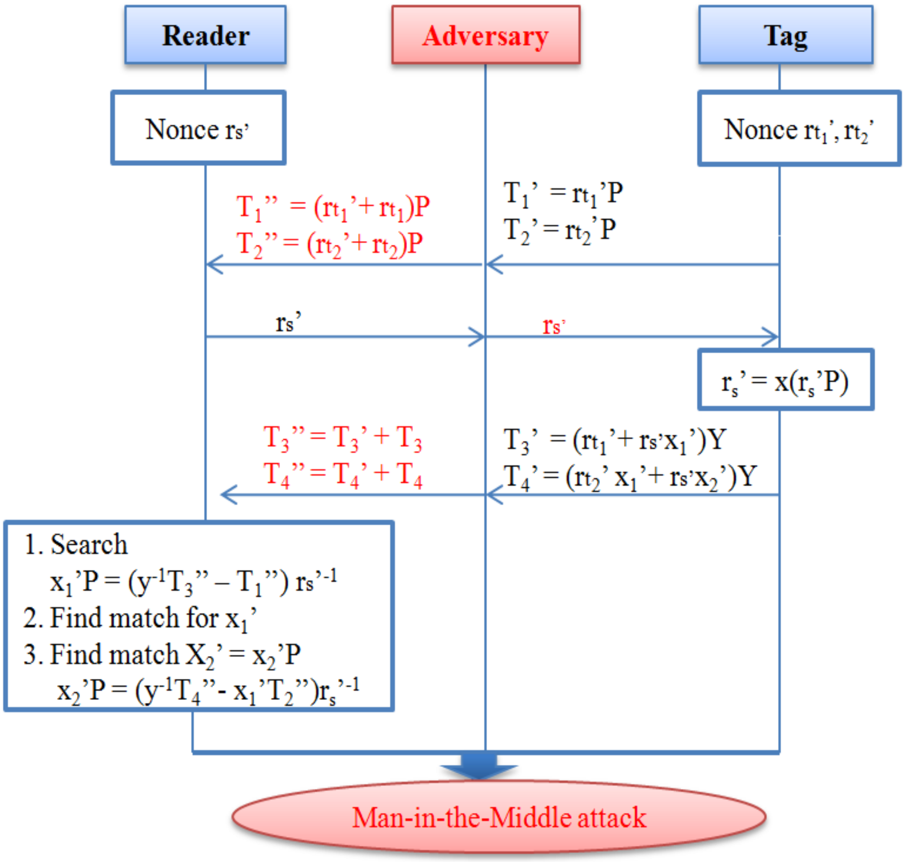


Table 8: Man-in-the-middle attack - Protocol 2, 3

Additional attack for protocol 2: even if the server adds a verification check whether r_s is equal to 0 and actually generated from a multiple of the order of P, an attack is still feasible. The adversary has to send the same r_s value in two different runs and subtracting them results in $r_s = 0$. Hence, the attacker then calculates the received values with the intention of acquiring valid T_1, T_2, T_3, T_4 and performing once again the first attack [79].

The ID and Password transfer scheme are narrow-weak privacy preserving. The protocols are vulnerable to adversaries that have wide and strong abilities. Untraceability is not preserved and tags can be tracked using a man-in-the-middle attack as described above [79].

4.4 Protocol 4

This protocol applies a challenge-response technique. The protocol aims to achieve mutual authentication, prevent desynchronization and untreaceability of tags. The following diagrams and accompanying explanations illustrate how this protocol fails to meet these requirements. (Original diagrams [83]).

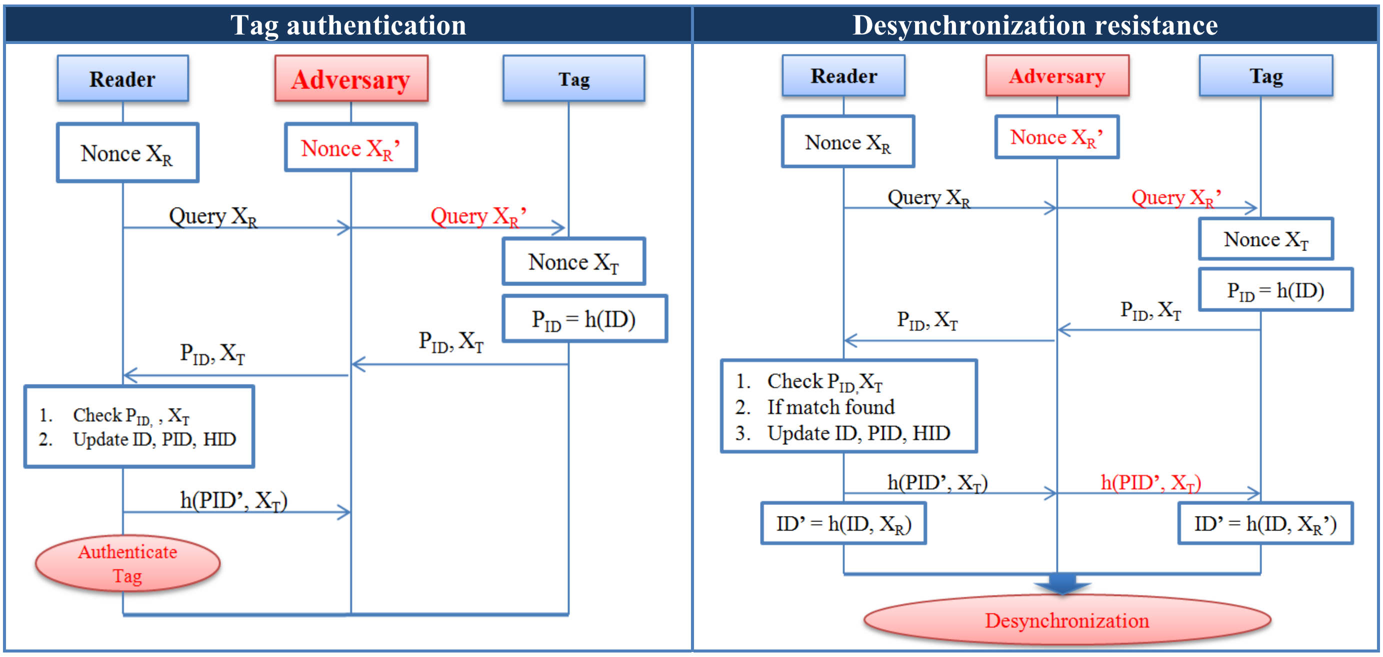


Table 9: tag authentication and desynchronization attacks

Authentication: According to Lowe's authentication definition [93], RFID protocols must accomplish recent aliveness otherwise, tag authentication fails. In this protocol, the challenge sent from the reader and the tag's response are independent, thus eliminating recent aliveness and allowing an adversary to impersonate a tag. This can only occur if the status is set to 0, which in a multi tag-reader environment the likeliness of several tags to fall in such a state is unavoidable. An adversary can query a tag (only when status = 0) and obtain *h(ID)* that can then authenticate the impersonating tag with the authorized reader [23, 83].

Untraceability: the protocol should prevent an adversary from determining the tag's status that can lead to distinguishability of tags. The authors claim that untraceability is established due to the freshness of the responses. However, an adversary can distinguish the tag's status by altering the random nonce generated by the tag and observing if the reader will accept or reject. If the status is 0, then the reader will accept even if the nonce is changed, else if the status is 1, the tag will reject upon realizing the change. Hence, the adversary can always guess the right tag with probability 1.

Desynchronization resistance: an adversary can perform a man-in-the-middle attack by altering the random nonce generated by the reader ($X_R \neq X_R$') and send the new challenge to the tag without modifying any other message. Because the status is set to 0,the reader does not check if the tag received the correct challenge. As a result, the execution continues 'normally' reaching the update phase were the reader updates h(ID, X_R) and the tag updates h(ID, X_R'). The protocol has no re-synchronization mechanism [23].

Forward secrecy: this property is partially preserved according to the status of the tag. If the tag's status is set to 0 then the response will consist of the tag's hashed ID. Otherwise, the tag replies with hash value of the ID and the random nonces. In the first situation, the messages are constant and forward privacy is unachievable.

Replay attacks: the tag always sends a generated random nonce, hence an attacker cannot perform a replay attack.

A tag performs approximately three hash operations and stores the ID plus an extra bit for the status. The overall estimated communication load in this protocol is approximately 4l. (l: length of an ID) [68].

4.5 Protocol 5

A-SRAC is the improved version of SRAC and includes the usage of random number generation from both the server and the tag. The authors claim that the protocol is secure against a wide number of attacks such as DoS attacks, forward security, spoofing attacks, replay attacks, location tracking and achieves strong privacy. However, according to Juels et al [41], formal definition on privacy, A-SRAC does not satisfy strong privacy if an adversary is capable of distinguishing tags. Other authors [43, 68], also examine weaknesses in the design:

DoS attacks: if an attacker blocks the last message sent to the tag to update its ID, it will result in the desynchronization of the system.

Location tracking: The protocol consists of three messages, exchanged between tag and server. In the case of a desynchronization attack when the last message is blocked or jammed, then the tag continuously emits the same hashed value, h(ID), thus the adversary can track the specific tag.

Indistinguishability: this protocol relies on random number generation, therefore an attacker has the ability to generate e.g. N_{R1} and send it to the tag. The tag will then constantly reply with the $h(ID||N_{R1})$ allowing the attacker to distinguishing the specific tag from the rest. Moreover, if this occurs, it also enhances the location tracking attack.

Ha et al. [64] claims that forward secrecy is partially preserved. This assumption is correct since the protocol uses randomly generated nonces but does not include them with the hashed ID. For this reason, replay attacks are prevented but in situations of desynchronization when a tag fails to update its ID, an attacker will be able to track the tag in future sessions.

Despite the protocol's failure to achieve important security requirements, ASRAC can be considered as a practical and useful authentication protocol that requires a logical amount of memory and computational resources. The tag stores the ID and the database stores $3x \cdot m$ (x: length of the ID, m: total number of tags). The total number of hash operations for the tag is three and for the database an average of four is estimated. The tag and reader perform a random number generation and $6x$ is the approximate overall communication load [43, 68].

4.6 Protocol 6

Designing RFID protocols for supply chains are challenging due to the nature of the system. The system consist of several partners with corresponding readers, that have to authenticate bulks of tags in a limited amount of time. In the mean time, the protocol has to offer unlinkability, supply chain visibility and authoritative access. Attacks on similar protocols [25, 26].

Reader authentication: as illustrated in the diagram, an attacker can obtain the random nonce N_R and send it to the tag, masquerading as a legitimate reader. The tag will then reply accordingly and the attacker will send the final message that will authenticate the reader to the tag and update the tag's pseudonym with a new tampered value [23].

Desynchronization resistance: if the reader fails to successfully authenticate, this leads to system desynchronization. The legitimate reader fails to send the random number used to update the tag's pseudonym while the tag carries on updating using the adversary's random number [23].

Untraceability: as seen in the analysis of the protocol the authors state that untraceability is not a requirement worth focusing on in such RFID applications. The tag does not authenticate itself since an attacker will always receive the same pseudonym update if he sends the same challenge N_R each time. Weak untraceability is not supported either by the protocol. An adversary can obtain the nonce

N_R, the hashed value of the nonce XORed the pseudonym, the values of R_1 and R_2 and the current pseudonym, i.e. $h(N_R \oplus a)$, a. With these values the attacker can now send $N_R' = N_R \oplus a$ and the tag will respond with $h(N_R' \oplus a')$ which is equivalent to $h(N_R \oplus a)$ [23].

Forward secrecy: the tag responses are not constant because randomly generated nonces are included with the hashed ID. Replay attacks for this reason are also prevented.

Below are two diagrams that reveal weakness in the design of the protocol. (Original diagram [23])

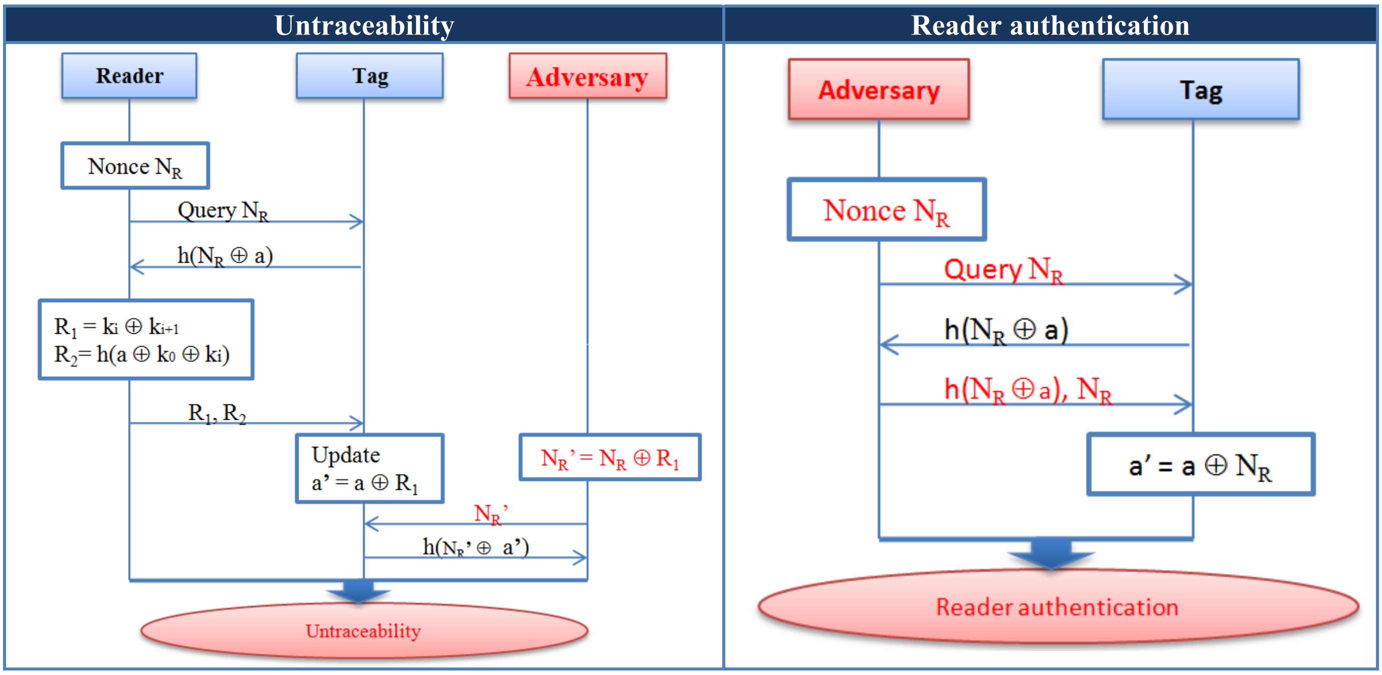


Table 10: Untraceability and reader authentication attacks

However, when considering the performance of the protocol's design, storage cost, and computation workload is kept low. Hash functions are mainly used and pseudonym *a* which is *k* XORed with *c*, is the only stored value on a tag.

4.7 Protocol 7

The HB protocol presented by Hopper and Blum [92] and the enhanced version HB+ by Juels et al [71], were both eventually proven weak security schemes. Although, both schemes are based on the LPN (Learning Parity with Noise) problem which is proven NP-hard, Gilbert et al. [78], demonstrates a man-in-the-middle attack against the HB+ protocol and Golebiewski et al. [80], presents a passive attack for the HB protocol which leads to an active attack against the HB+ protocol.

Golebiewski's attack consists of an algorithm able to break an 80-bit HB with noise parameter 0.25 and keys with a maximum length of 96 bits. In order to proceed with the attack, monitoring two successful runs and collecting a number of observations must be separated into two divisions. The one division is for finding the secret key *x* and the other is to select random candidates for the secret *x*. The authors carry out a procedure using a type of brute force attack, that runs through a set of challenge-responses obtained in the two protocol executions and eliminating possible keys *x* until the identify the definite secret value [80].

According to Golebiewski's attack, the HB protocol was broken in several hours on a home computer for key size $n = 96$ and 154 bit, noise parameter $h = 0.25$ and 0.125. An attacker has to eavesdrop and obtain O(n) challenge-response pairs with the help of their algorithm which requires a sample of exponential numbers. For further details on the attack and algorithms [80].

Gilbert et al. [78], present a theoretical attack were an adversary performing a man-in-the-middle attack during an authentication process, manipulates the challenge messages sent from a valid reader to a valid tag and monitors the verification results. In Gilbert's attack, if the adversary successfully reveals the secret *x*, then the following consequences are: valid tag impersonation, revealing the second secret *y* and weakening the secrecy of the tag's secret identity and thus reducing privacy.

The diagram below and the accompanying description, illustrate the attack. (Original diagram [78])

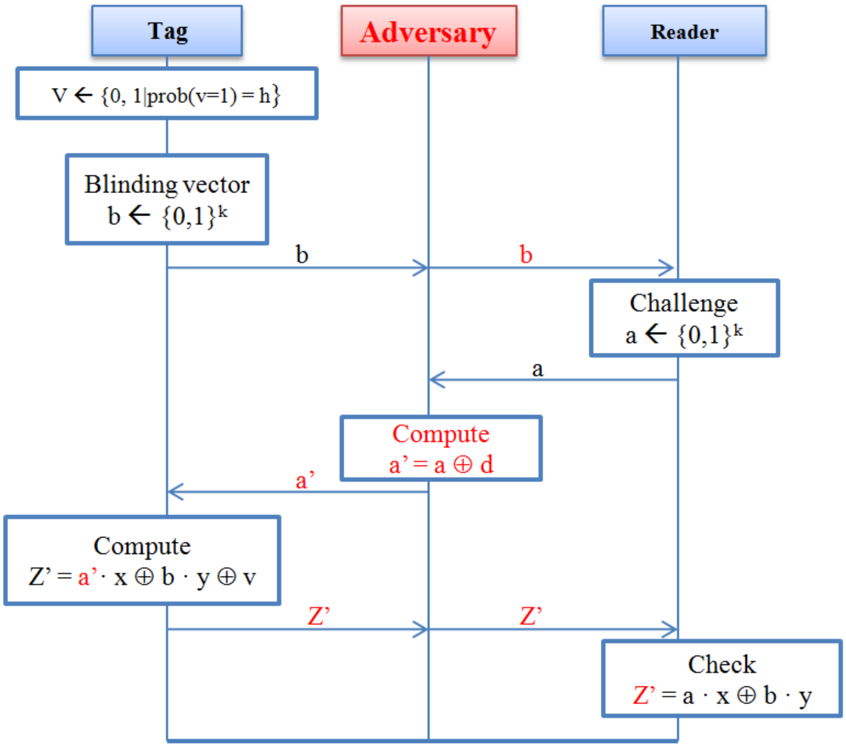


Table 11: Attack on the HB+ Protocol

Recalling the HB+ protocol terminology, the tag and the reader both hold two secrets *x* and *y*. A blinding vector *b* is generated by the reader and a value *v* is a noise bit equal to 1 with a probability *h* ← *[0,1/2]* generated by the tag. The attacker generates a constant k-bit vector *d* that is XORed with the challenge *a* for all the runs of the protocol. The aim of the adversary is to reveal the complete secret *x* and the attacker can achieve that with the results of the authentication. If the reader accepts or rejects the authentication, the result reveals a single bit of the secret information, i.e. accept $d \cdot x = 0$, reject $d \cdot x = 1$. The secret key *x* requires *k* rounds of the same attack to be repeated in order to reveal all the secret bits.

Forward secrecy: the tag responses are not constant because randomly generated nonces are included in the message Z. Replay attacks for this reason are also prevented.

Even though the HB and HB+ protocols are susceptible to passive and active attacks, the principles of the protocols can be used as a background idea for designing a shared-key authentications protocol for low-cost RFID tags. Reason are that the keys and procedures used in the protocols are efficient for low-cost RFID requirements and the most complex operations are XORs and AND operations.

4.8 Protocol 8

Recalling from chapter 3, this protocol is a randomized hash-based, access control scheme and claims to prevent a number of possible attacks such as desynchronization, cloning by eavesdropping spoofing and replay. Untraceability, scalability, in some cases ownership transfer and support of multi-tag-reader environment are also properties the protocol is supposedly capable of attaining. According to the design analysis of the protocol, timestamps, hash functions and access passwords (secret key) are the main operations of the scheme.

Several feasible attacks result in the protocol's failure to achieve untraceability, tag authentication and desynchronization resistance [23].

Desynchronization resistance: similar with protocol 5, if the last message sent from the reader to the tag is blocked this will cause desynchronization during the updating process, i.e. reader successfully updates while tag fails to receive the necessary information in order to update as well [23].

Recent aliveness: it is important for protocols to achieve recent aliveness, i.e. the tag's response and the reader's challenge must be dependent otherwise, replay attacks can occur, and forward privacy can be compromised.

Forward secrecy: although the reader generates a timestamp that is different for every session and forwards it to the tag, the tag then fails to include the timestamp with the hashed ID in order to prevent the messages from being constant.

Tag authentication: the reader's challenge and the tag's response are independent. The reader is supposedly authenticated by the tag after the first message, *h(k,TS), TS* has been received and checked. The reader authenticates the tag once the second message, *h(ID)* has been received and checked. In such protocol designs, an adversary has the ability to impersonate a tag by replying to the reader's challenge. Hence, the reader might be authenticated but the tag's authentication to the reader is omitted [23].

Untraceability: one of the key features the authors of the protocol focused on. Tags are supposedly untraceable since their responses to reader queries change for every valid session. However, an adversary can trace desynchronized tags that do not complete a valid session. This is due to the lack of desynchronization resistance that results in authentication failure and early termination of the session. Other consequences occur when the tag is in a desynchronized state and constantly replies with the same message since it fails to update its ID and timestamp. The techniques of using timestamps although they are an efficient way of reducing cost as alternatives to timers, adversaries can easily determine the difference in time between two desynchronized tags [23].

Scalability according to the authors is O(1) and O(N) for multi-tag-reader environment. An average of four hash functions in a protocol is common but can be reduced if an alternative to verifying the reader's message is used. Timestamps if used correctly i.e. included in the tag's hashing of messages, are a practical solution for preventing replay attacks.

4.9 Protocol 9

Kang et al. [73], highlight location privacy as an essential property for RFID systems. They claim that DoS and spoofing prevention are achievable with their scheme due to the threshold counter and constant identical responses. The counter is supposed to increase each time it receives a message. If an attacker insists on sending messages then the counter will eventually reach the maximum value and terminate the session. The technique of constant identical responses ignores the attacker's request.

Both methods prevent the attacker from tampering with the system. The protocol is however, vulnerable to several attacks and does not provide the claimed privacy.

Tag Anonymity: According to [23], it is possible to determine the tag's ID. The messages transmitted during the protocol execution leak a certain amount of bits. i.e. HID $\leftarrow$ h(ID|| R_2), $NR_1 \leftarrow$ (ID $\oplus$ R_1), ID $\oplus$ R_3 , $R_1 + R_3$. An attacker can perform a brute force search once he has gained enough bits of an ID and compare them with the first message sent from the tag, h(ID||R_1), revealing the ID from a set of possible ID's. In order for such an attack to be mountable and the tag's ID to be completely revealed, it requires the observation of several protocol runs. Though obtaining the full bits of an ID can take up further effort because the protocol on the last message XORs the ID with both random nonces. The attacker once attained the tag's ID he can then impersonate the tag or reader. Due to the possibility of an attacker to reveal a tag's ID, the consequences include breaking both the reader's authentication and untraceability. For further details on the attack [23].

Desynchronization prevention: is also affected since an attacker is able to block the last message that prevents a reader from updating the tags new ID and secret key k.

Information leakage: if the attacker can successfully reveal the tag's ID then the information stored on the server that correspond to the specific tag is revealed since an attacker is able to impersonate a valid tag.

Replay attacks: are prevented since the random nonces are used and refreshed for every new session.

Forward secrecy: the protocol preserves this privacy property because the messages sent by the reader are not constant. Due to the random nonces, that are either hashed or XORed with the tag response, it results in a fresh message that is only valid for that current session.

4.10 Protocol 10

The SPA protocol, as described in chapter 3, uses a tree based approach and a novel key-updating design. However, it is clear that such designs were a single tag holds a number of keys, large amounts of memory are essential. The authors also claim that their new key-updating mechanism solves the main disadvantage of tree-based approaches by eliminating the tag-compromising attack. When secret information is shared amongst tags, there is a possibility that if a single tag secret is exposed, then other tags sharing the same secret will be vulnerable to attacks. Similar attacks on protocol [22].

According to Daou et al. [43], this scheme resolves most of the security issues except from compromising attacks and man-in-the-middle attack. An adversary has the ability to masquerade as a legitimate reader and interact with the tag long enough to capture the necessary information that can lead to other attacks such as tag cloning or altering tag information. Compromising attack is supposedly prevented by the key-updating method, introduce as a novel design. Nevertheless, an adversary can still perform the attack under certain circumstances and successfully tampering with one tag will expose a set of common shared keys.

This protocol attempts to employ important security properties such as tag anonymity, replay attack, forward secrecy, user data confidentiality [43]. When considering attack preventions, authors must take into account that if one type of attack is mountable then others can follow. For example, this protocol is unsecure against compromising attacks, if tag secrets are exposed then cloning attacks are possible.

User data confidentiality is also worth examining since, tag-compromising attacks are feasible, the attacker can gain access to a tag's key and access confidential information.

Forward secrecy is preserved, since efficient updates take place erasing any old communication messages. However, if the system fails to update the keys due to a DoS (Desynchronization) attack, the tag will respond with constant messages. This can lead to untraceability.

Furthermore, Daou et al [43], proposes an improvement that enhances the SPA protocol to achieve DoS attack and compromising attack prevention. The system can implement a session identifier, e.g. counter, that will force a tag to compare its hashed session number with the readers request session number and if they match, the tag will then authenticate the reader. Otherwise, the session will halt. To prevent compromising attacks, the session number can be used during the hashing of keys [43].

The protocol approximately requires an average of nine hash operations. The overall estimated communication load is can increase if the systems involves a large number of tags. The authentication is divided into three steps with scalability of O(1). [43, 48, 68].

4.11 Protocol 11

Dimitriou's lightweight protocol [75], introduces a counter that increments each time a successful mutual authentication is complete. In addition, a secret key shared between tags and the back end-server is supposedly renewed after each session. The reader does not store any previous tag IDs once they have been updated, and thus the system fails re-synchronization in a DoS attack. According to [68, 74] this protocol is vulnerable to the following attacks:

Location tracking: this occurs when the system is in a desynchronization state and the attacker sends a request from the reader to the tag. The protocol does not prevent an attacker from sending an adequate amount of queries affecting the indistinguishability of tags. This is due to the unchanged responses from the tag, h(ID) [74].

DoS attack: for the authentication process to be complete, a third message is sent from the reader to the tag. An attacker able of blocking the last message, results in the system entering a desynchronization state. This will result in the reader updating the tag's new ID and secret key, while the tag fails to update either of the two values. The consequence of desynchronization prevents the system from re-entering a synchronization state and the targeted tag is futile.

Forward secrecy: the tag responds with three messages and one is the hashed ID of the tag, h(ID). In order to preserve forward secrecy, the messages must not be constant. The protocol uses randomly generated number for each session but fails to include them with the hashing of the ID. Thus, replay attacks are prevented but an attacker is able to track the tag in future executions of the protocol.

The protocol is secure against cloning attacks, it preserves tag anonymity and user data confidentiality. Reasons are that although the messages of the tag might be constant, and a DoS attack is mountable, the tags ID and secret keys are hashed and then sent to the reader. The average amount of hash operations the reader and tag perform are approximately four and the overall communication load of the system is also approximately 5l (l: the length of an ID) and authentication efficiency O(logN) [48, 68].

4.12 Protocol 12

Juel's "minimalist" cryptography protocol is based on a challenge-response, one-time pad security scheme with pseudonym updates and XOR operations. The protocol has been cited by several other authors either to identify the limitations of the protocol or to propose an updated version [41, 42, 58, 74, 75].

The main limitation of the protocols is the extensive memory requirements. The protocol consists of four communication messages and it is essential for keys and pads to be updated with new values, i.e. secrets [75]. Due to limited memory resources on the tag, a large set of pseudonyms cannot be stored. As a result, after several authentication sessions, the tag will have used up all the stored pseudonyms and will require an update in order to prevent security attacks. The protocol has two options. Either to reuse pads that undermines the security level of the system or to replace pads through an out-of-band channel that if performed frequently, limits the practicality of the scheme. [94]. In [58], the authors mention that this scheme requires a new RFID tag design and the security is weakened due to the lack of memory resources that cause pad reuse. Furthermore, in [defining strong priacy] the authors re-highlight the privacy definition of the original author claiming unsuitability in a man-in-the-middle attack scenario that has a limited number of queries, resulting in a weak privacy preserving protocol. Finally, even though the limitation of the scheme's practicality is underlined in many papers, it is a good initiative for low-cost tags that cannot perform symmetric-key cryptography [75]. The overall authentication efficiency requires approximately O(logN) [48].

4.13 Comparison

The table below illustrates the general comparison of the achieved security properties for the analysed protocols in this section. The comparison was also structured with the aid of several other papers that investigate and reveal security weaknesses in low-cost RFID authentication protocols [48 - 55]. The protocols that are marked with either ✓(Δ) or X(Δ) represent controversial claims in the literature review. Δ symbolizes contradiction and X or ✓ the most rational assumption.

Properties vs. Protocols	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
Forward secrecy	✓	X*	X	P	P(Δ)	✓	✓	X(Δ)	✓(Δ)	P	X	✓
Cloning Attack Prevention	X	✓	✓	P	✓	✓	✓	X (Δ)	X	✓	✓	X
DoS Attack Prevention	✓	✓	✓	X	X	X	✓	X (Δ)	X	✓	X	X
Replay Attack Prevention	✓	✓	X	✓	✓	✓	✓	✓	✓	✓	✓	✓
Tag Anonymity	✓	X	X	✓	✓	✓	X	✓	X (Δ)	✓	✓	✓
Untraceability	X	X	X	X	X(Δ)	X (Δ)	X	X	X (Δ)	X	X	X
User data confidentiality	✓	X	X	✓	✓	✓	X	✓	X	✓	✓	✓

✓: prevents attack
X: cannot prevent attack
Δ: controversial in literature review
P: partially prevent such attacks (under certain circumstances)
*: P2 consists of 6 protocols, 4 and 6 do not have this property

4.14 Summary

This section evaluated several proposed RFID protocols and assessed their privacy and security properties. As seen from the above comparison, the protocols fail to achieve a number of security properties, which result in privacy issues arising within an RFID system. The most significant conclusion worth emphasizing on is the fact that all the above protocols are based on XOR operations, hash functions, symmetric key techniques, passwords, updating secret via the database and other similar methods. However, even though these techniques might remain within the boundaries of low-cost, it is obvious that they are not enough to secure an RFID tag and prevent attacks. Hence, the following chapters introduce a new RFID protocol based on ECIES (Elliptic-Curve Integrated Encryption Scheme) and Vaudenay's privacy model [56]. The proposed protocol achieves a higher level of security and privacy in comparison to those of the prior art reviewed in this chapter.

5. Description of an RFID Privacy model and other essential concepts

This section describes the background concepts that will form the final design and proposal of the RFID protocol. This section investigates Vaudenay's privacy model [56], Elliptic-Curve security processors [44], and AES implementations for passive RFID tags [45]. These three subsections are linked with the accompanying design principles and form the fundamentals for introducing the protocol. The reason for describing AES implementations and EC processors for RFID tags, is to augment the evidence that the design of the proposed protocol is efficient in both its security and performance requirements.

5.1 RFID Privacy models

The author's paper on privacy models is analysed and presented in this section with the intention of demonstrating the privacy models that RFID system focus on, the various types of adversaries, both the limitations and advantages of an adversary, and finally the highlighted privacy model – narrow-strong and forward privacy model that uses public key cryptography (PKC).

Vaudenay describes several RFID privacy models and sorts them in a hieratical way, from the weakest up to the strongest. The author includes both security and privacy analysis and proves that the only way to achieve the strongest possible level of privacy in an RFID system is to use key agreement i.e. public key cryptography techniques, which although considered costly, is unavoidable [56]. The author also takes into consideration Ohkubo et al. [57, 85] RFID protocols and adjusts it to one of the privacy models proving that the new version achieves narrow-destructive privacy. A simple challenge-response protocol, similar to the protocols described in previous sections, is proven by the authors to provide weak privacy. EC-RAC protocol [67], presented their protocol based on Vadenay's narrow-strong privacy model, however as seen in the previous section, the authors fail to take into consideration essential PKC points and thus the protocol is proven insecure. The terminology is described in this chapter [56].

This section enhances the projects following section, the proposal of the new RFID protocol design. Below are some of the highlighted and important points the author mentions[7] [56]:

- Strong privacy is unachievable.
- The strongest level of privacy is narrow-strong.
- Powerful adversaries have the ability to control all communications and interfere with the system.
- Narrow-strong and forward privacy model are secure against CCA attacks and can be employed into a key agreement protocol that exploits public key cryptography.
- The author focuses on security issues that involve tags and not readers.
- The privacy models are concerned with cheap tags that have the following specifications: passive, not tamper proof, no batteries, operate only when interrogated by a reader and for a short time, restricted to limited distance, limited memory and computational abilities, perform basic cryptographic calculations e.g. pseudorandom generation, hash function, symmetric encryption.
- Security is responsible of the soundness of tag authentication whereas privacy defines the level of adequacy that restricts adversaries from tracking, identifying and distinguishing tags

[7] Note: the terminology and expressions used in this section is the same as the ones used by the author.

- Identification vs. authentication protocols: if the tag is legitimate the former outputs the private value which is the tag's ID or else ∅. The later outputs the private value 1 or 0, if the tag is legitimate or not, respectively.

The author presents several privacy restrictions models. The following hierarchy description of the models starts from the weakest up to the strongest privacy model [56]:

1. Weakest model: prevents tag corruption.
2. Forward-privacy model (same as backward privacy): Permits tag corruptions under certain circumstances such as, at the end of the attack were no additional operations are allowed.
3. Destructive-privacy model: tag corruption can occur at any time but once a tag has been tampered with, it can no longer function appropriately.
4. Strong-privacy model: an adversary can corrupt tags at any time and has an additional ability to restore the tag back to its purpose allowing tracking attacks to follow.
5. Accompanying with these models are two types of adversaries that can either access the reader verification results or not. The former is called a "wider" adversary that can access the output i.e. if a reader accept or reject a legitimate tag from side channel. The later is a "narrow" adversary, which cannot access verification outputs.

An RFID scheme consists of two phases. First, an initialization phase initializes the long-term keys i.e. produces the cryptographic keys. The author also mentions a setup phase that is responsible for distributing the public key (PK) and a secret key (SK). Both keys must depend on the system's security parameter. The setup phase uses a polynomial-time algorithm used to setup the tags. The algorithm outputs a secret key K that depends on the input value ID, and the initial state of the tag S. The RFID scheme outputs ID or ∅ according to the verification result from the reader. The former occurs when the tag has successfully identified a legitimate tag and the later is for invalid tags [56].

The author divides tags into two states and any tag can move between these states. Drawn tags are the ones that adversaries can interact with and track. They also are identified as "virtual tags" (temporary reference value e.g. vt). Free tags are out of an adversary's reach. Adversaries are separated according to the oracles they can access and the results they can obtain. The adversary has access to the following oracles [56]:

1. CreateT(ID,b): this oracle interacts with SetT$_{pk}$(ID) and creates then sets up free tags with unique IDs that depend on the bit b. If the b=1, the tag is legitimate else if b=0, the tags do not belong to the system.
2. SelectTag(p_dstr): A set of free tags, (vt$_1$,b$_1$,....,v$_i$, b$_i$), are randomly selected by the system according to the probability distribution (p_dsrt). The value b indicates if the tag is legitimate (b=1) or not (b=0). The output of this oracle is a set of drawn virtual tags with unique IDs and bits (b1... b$_i$).
3. Free(vt): if a tag is set to a drawn state, then by calling this oracle the specific tag is free.
4. Start → p: this oracle starts a new instance with the reader.
5. MsgReader(m,p) → m' or MsgTag(m,vt) →m': this oracle is responsible for sending messages m at an instance p from a reader to a tag and vice versa. The representation of the replies is m'. This oracle also use another oracle Transcript(vt) →(transc,p) responsible for returning the transcript off all the communication messages send to and from the reader and a virtual tag.
6. Result(p): the oracle stores the verification result of the instance p once it has been completed. If the result is 1 then the output is valid and returns the tag's ID else it returns 0.
7. Cor(vt)→S: when this oracle is called, the tag's status is returned and the virtual tag, vt, is corrupted, e.g. no longer functional.

The privacy model takes into consideration several types of adversaries with varying capabilities e.g. Strong ≥ Destructive ≥ Forward ≥ Weak. Below is the summary of the each adversary [56].

1. Strong: this is the strongest form of adversary. They have access to all the oracles and can use them accordingly.
2. Destructive: these adversaries cannot use a virtual tag once they have called the Cor(vt) oracle since it destroys the specified tag.
3. Forward: Queries to the Cor(vt) oracle can only take place when an attack reaches the final stage to complete.
4. Weak: cannot call the Cor(vt) oracle for virtual tag corruption.
5. Narrow: cannot call the Result oracle in order to obtain the outputs of protocol instances.

When considering the issues of privacy in an RFID protocol, the author stresses on two key properties, anonymity, and untraceability. Anonymity is concerned with keeping the tag's ID secret and untraceability tries to preserve indistinguishability between tags. The goal of an adversary is to reduce privacy by outputting a set of virtual tags and ascertain a link between their IDs. The author states that an RFID system provides strong privacy if there is no significant strong adversary [56].

According to the author [56], an adversary is linked with a blinder that is a polynomial time algorithm, which, has no access to the secret keys but can acquire the same messages as the adversary. It is responsible for simulating several oracles for the adversary e.g. Start, MsgReader, MsgTag, and Result. If the exists an adversary A and a blinder adversary AB, the protocol preserves privacy if the probability of A subtracted from the probability of AB winning, is negligible.

In order to determine the security level of an RFID scheme, the author challenges the models against strong adversaries. The general procedure is as follows. The adversary first has to select a tag from the set of drawn tags, which can be either legitimate or illegitimate, and executes instances of the model. An adversary wins if at the end of at least one instance p, he successfully identifies a legitimate tag. The author adds other security challenges such as during an attack, a tag's ID cannot execute the whole instance p or the corruption of a tag's ID was prevented (i.e. identified the tag without corrupting it) or an execution of the protocol successfully ends during an ongoing attack. If the scheme can prevent the above scenarios then the protocol is said to be sound and complete if no adversary can identify a legitimate tag [56].

The narrow-strong and forward privacy model according to the author, is the model that achieves the highest level of security for low-cost RFID tags and the design must be based on Public Key Cryptography (PKC). The proposed protocol in the following section, adopts this privacy model as its cornerstone in order to acquire the same level of privacy for the RFID protocol design. The model works as follows. The tag is responsible to encrypting the stored values i.e. ID, secret key and the received random value from the reader. The reader decrypts the values and checks if they match any stored values on the database. If a match is found, the reader then identifies the tag and outputs the tag's ID or an error value accordingly [56].

Vaudenay initially explains what a key agreement protocol consist of, why it is considered suitable for an RFID privacy scheme, and how it associates with PKC. A key agreement protocol consists of two parties, i.e. tag and the reader, and both share a public value linked with the security parameter of the system. They interact, exchanging messages and finally output the corresponding value. Key agreement protocols can be characterized as an interactive protocol. The probability of an adversary that has access to the common values and the transcript of the model, determines the level of security. If the chance of guessing the secret value is more than 0.5% probability plus a negligible advantage, the protocol is considered insecure. The author with the help of an adversarial challenge game against the system illustrates that a key agreement scheme is secure since an adversary cannot obtain any information on any of the tags ID. Furthermore, an RFID scheme based only on pseudorandom functions, symmetric primitives, hash functions, digital signatures etc cannot achieve narrow-strong privacy [56].

The author proves that the public-key cryptosystem achieves narrow-strong privacy if it is IND-CPA secure and forward privacy if IND-CCA secure. There is a narrow-strong adversary A which is associated with a blinder B and A does not possess any further advantages or is more powerful than

AB. *B* sends an input *x* to the MsgReader oracle and obtains the encryption *c* of a random value *n* that has the same length as *ID||k||x*. In the mean time, there exists a simulator *S* (polynomial algorithm) for the *AB* attack. The IND-CPA[8] game runs as follows. *S* gains possession of the public key *PK* and computes m_0 and m_1 ($m_0 = ID||k||x$, $m_1 = n$) and forwards them to the challenger. The challenger will then respond with an encryption C_b *(b ϵ {0,1})* of either m_0 or m_1. When simulator completes its executions, it then verifies the status of *AB*, if *AB* won then *S=0*, if not *S=1*. Finally, the probability of *A* winning subtracted from the probability of *AB* winning is negligible. Hence, the protocol is IND-CPA secure and narrow-strong privacy preserving [56].

The protocol is also forward privacy preserving since it is secure against CCA attacks. The CCA game is similar to the CPA above. There exists a blinder *B*, a simulator *S*, an adversary *A*, all used in the *AB* attack. The simulator ends the game by determining if the adversary has guessed the correct message by sending the ciphertext of *A* to a decryption oracle. The chance of the adversary *A* subtracted from the chance of blinded-adversary AB winning the CCA game is negligible. In situations where a tag is no longer needed in the system and the reader sustains securely its information on the tag, an adversary or any other third illegitimate party cannot identify the tag. In order to identify a tag, its secret key *K* must first be determined. This can only happen with negligible probability (2^{-k}) [56].

5.2 Using Elliptic-Curve-Based Security Processor for a PKC RFID Protocol

According to Lee et al. [44], a public-key cryptosystem is the *only* way to accomplish privacy, strong security, reasonable performance and scalability when considering the design of an RFID protocol. Therefore, an Elliptic Curve-based cryptosystem is the most practical approach due to its small key size and desirable performance. The authors present an EC processor that combings techniques to reduce the number of registers and perform EC scalar multiplications and general modular arithmetic, addition and multiplication. The proposed protocol design uses ECIES based on Vaudenay's [56] privacy model and with the help of Lee et al. [44] findings, an efficient Elliptic-Curve processor that requires 0.13ms CMOS technology and gate area approximately 12.5Kgates can be applied were needed for this scheme [44].

The authors identify several important statements and techniques for designing EC processors. Below is a brief mention of these statements:

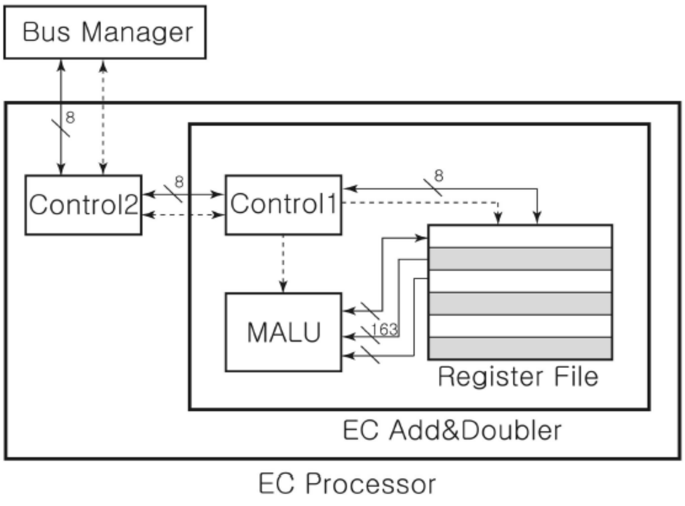
When considering the design of an EC processor architecture for RFID tags, the total number of register and the complexity of the register files play a significant role. Registers are used to store field operands and more than half the gate area size is taken up by the registers. For this reason, the authors focus on reducing the number of registers that also affects the gate area size significantly [44].

The authors state that applications implementing EC systems have a choice between the two most common types of curves e.g. GF(p) or GF(2^n). Both types offer an equal level of security except the authors prefer the binary fields due to its implementation flexibility. The binary method is the basic scalar multiplication algorithm that the authors expand on and produce there three architectural versions [44].

[8] IND-CPA: Indistinguishability under Chosen Plaintext Attack. The game runs as follows. For any polynomial time adversary that receives the public key PK, he then computes two messages m_0 and m_1 and submits both to the challenger. The challenger will then chose at random (b ← {0,1}) one of the two messages to encrypt and return the ciphertext to the adversary. Next, the adversary computes the value d, which is his guess and submits it. The adversary wins if the guess d=b, meaning the adversary guessed which of the two message was encrypted. A protocol is considered secure if an adversary with probability less than 0.5% plus a negligible advantage can guess the value d.
IND-CCA: Indistinguishability under Chosen Ciphertext Attack. The game between the adversary and the challenger is the same as the CPA, the main difference is that the adversary has access to a decryption oracle and can decrypt any ciphertext he wishes except from the challenge ciphertext.

The designs are based on the Montgomery ladder with the Lòpez- Dahab algorithm and a unit responsible for arithmetic field operation called the Modular Arithmetic Logic Unit (MALU). The storage of the Y-coordinate is not necessary and MALU architecture is considered a compact architecture, thus the reason of choice. Although, the Lòpez- Dahab algorithm and the new algorithms require the same amount of computational workload the authors successfully reduce the number of registers. Other works similar to these authors, present the Lòpez- Dahab algorithm requiring nine registers were three are for the MALU and the rest are for the Montgomery algorithm. The various techniques the authors apply i.e. reuse of registers, reduce the total amount of registers from nine to six. The six registers are for the following field operands, X_1, X_2, Z_1, Z_2 and intermediate values: T_1, T_2. The scalar values are 163bits and an additional of five extra guard bits result the scalar values to be 168bits (21bytes.) in an 8-bit controller [44].

For both the modular addition algorithm and modular multiplication algorithm the authors present enhanced versions that lead to the resourceful results. Various other techniques the authors apply are to increase the computational workload that reduces both the registers and the gate area. Registers are also used for two reasons, to store field operands as well as intermediate values. The reduction of the registers can be found in the efficient design of the MALU that allows the reuse of registers during the Montgomery ladder algorithm. These techniques and the arrangement of the six register in a single register file, decreases the gate area by 30%. However, an important point that must be avoided, is the concurrent updating of registers that can cause an issue if power consumption reaches high levels. These designs update the maximum of four register each time [44]. (Original diagram and description [44]).

EC Processor Diagram	Description
	Control 1: responsible for receiving the EC parameters and outputs the EC scalar multiplication via Control 2. Control 2: is a transport module for the data sent to and from Control 1. It obtains the key through the bus manager. MALU: performs the field operations (multiplications, squaring, additions). Register file: considered the most significant part in the EC processor architecture. It is where the registers are kept and managed and it occupies the largest area on the hardware. The registers store the field operands. Bus manager: is responsible for transferring data to and from the EC processor to the rest of the RFID tag.

Finally, the authors present an efficient and fast way to compute the points, suitable for an RFID protocol by presenting three different types of the ECP architecture design and comparing them in terms of memory, power consumption, and number of cycles.

Briefly, the main differences between the three versions are:

Version 1: considered the most compact solution. It requires the most number of cycles in order to complete the Schnorr protocol and the least gate area. This version does not store the register for the EC base point and therefore has to reload the X-coordinate value of the base EC point at every execution of the algorithm.

Version 2: The X-coordinate value for the base of the EC point is stored in Version 2 and compared with Version 1 it requires less access to the ROM memory. The number of cycles and gate area size is somewhere in between version 1 and 3.
Version 3: demands the maximum gate area size, and the minimum number of cycles. An additional feature in this version, is the extra randomly accessible register file which causes the increases of the gate area but reduces the number of cycles.

The total power consumption for passive tags is a critical factor. Another important factor, that changes the gate area and the number of cycles, is the digit size that is also related to the power consumption. The authors present graphs that illustrate the effects of these factors when they are either increased or reduced. For example, when the digit size is increased then the power consumption is lower but leakage power increases. In addition, as the gate area size increases the number of cycles required to complete the Schnorr protocol decreases. There is a significant trade-off between the three version and according to the desired digit size, each version satisfies different conditions [44].

While scalability is a critical requirement for RFID systems, the authors suggest that to complete one EC scalar multiplication it requires around 250µs (two EC scalar multiplication = 500µs). Although, this is considered a reasonable response, in multi-tag environments this will cause an issue. Hence, the authors recommend that multiple tags should be processed in parallel with a multiple-access protocol [44].

As a final point, the authors take into consideration the ISO 18000-3 (13.56mHz) that in order to reach an operating range of 1m, the maximum power consumption is 15 µW at 1.5V. The finding of this paper concludes that, the power consumption can be limited by reducing the number of cycles, the gate area and increasing the digit size. Comparing the three versions, the accomplished power consumption is 10µW, which is the most practical solution for passive tags. Therefore, schemes that require the calculation of EC points can take into consideration the EC processor described above which is feasible to implement in a low-cost RFID tag. The proposed protocol in the next section analyses the performance analysis, since ECIES involves the calculation of EC points [44].

5.3 Advanced Encryption Standard as a cryptographic primitive for RFID protocols

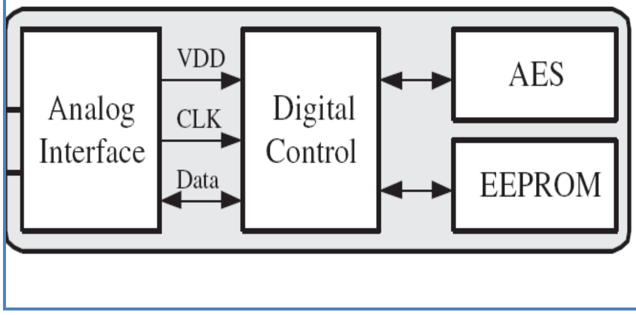
The proposed protocol in the following section applies AES encryption and therefore this section presents a background overview of the AES encryption for low-cost tags. This section describes how the AES encryption module are implemented on RFID tags.

AES operates on a block of data with a fixed size of 128-bits. The block of data is arranged in a matrix i.e. four rows by four array of bytes. The key lengths are 128, 192 or 256 bits and there are 10,12 and 14 round functions depending on the key size that cause the block of data to be altered. Efficient implementations are possible on 8-bit,32 bit, 64-bit and 128-bit platforms.

AES main operations of a round function are:
AddRoundKey: this is where the use of the keys are combined with the block of data by adding additional bytes and XORs the state and the round key.
SubBytes: also known as the S-Box operation, it is responsible for substituting each byte of the matrix (i.e. state) in a non-linear form.
ShiftRows: the rows of the state are shifted by a number of offsets.
MixColumn: this operation merges the columns of the state, i.e. four bytes into one column.

A number of AES implementations for RFID protocols have been investigated [46, 47, 84], and the most efficient solution is presented by Feldhofer et al. [45]. The author provides sufficient guidelines for an AES encryption architecture for RFID protocols. The description below demonstrates an example that analyses the prerequisites when considering AES as the symmetric encryption for the proposed RFID protocol.

The author takes into account the ISO/IEC 18000 standard with air interface communication at 13.56 MHz [45]. However, the RFID tag is forced to reduce the internal clock frequency to 100kHz and is restricted to reply to a response within 320 µs or else is must remain silent. Thus, a new proposal to the AES architecture implementation was needed given that, a response within 320 µs at frequency of 100 kHz does not fulfil the demands for AES encryption on a passive RFID tag. (Original diagram and description [45])

RFID tag architecture including an AES module	Description
 Analog Interface VDD CLK Data Digital Control AES EEPROM	Analog Interface: responsible for modulation, demodulation, and transferring the power from the reader and to the tag. Digital Control: allows reader/write to the EEPROM and AES module. Also, responsible for executing the protocol, communicates with the reader, and includes anti-collision mechanism. EEPROM: stores permanently the tags secret information i.e. ID, secret key etc. AES: designed for low power consumption and limited die-size.

According to Feldhofer et al.[45], they present a novel hardware design for AES that requires 1000 clock cycles and power consumption less than 9µA on a 0.35 µm CMOS process[9]. The protocol focuses only on the AES encryption. The author states that this allows a reduced amount of computational complexity and minimizes the space on the circuit.

A 32-bit AES implementation on a RFID tag is impossible so it must be reduced to an 8-bit architecture. The number of S-Boxes can be reduced from four to one and the power consumption is significantly reduced. However, the encryption process requires an increased amount of clock cycles. Another important point is the consumption that the digital controller consumes which is around 5mA subtracted from the total available power consumption for an RFID tag, approximately 20mA, leaving the AES module to operate on 15 mA. In addition, the S-Box that executes the SubBytes operation entails the largest part of the AES architecture. Therefore, the S-Boxes and the MixColumn are the two operations that undergo the most adjustments. When the number of S-Boxes increases, the encryption requires less clock cycles and since there are options to implementing an AES S-Box, the authors prefer to calculate the substitution values using combinational logic and excluding the decryption circuitry. The usual size of the MixColumns operation is reduced to a quarter. The AES encryption process does not require any intermediate memory and an "on-the-fly" key schedule calculates the round keys on the spot [45].

Finally, the author compares his finding with Mangard et al. [46] and Verbauwhede et al. [47] and demonstrates that his approach for implementing the AES encryption on an 8-bit platform accomplish the best results. For this reason, the implementation of this paper is recommended for the proposed protocol design. A more analytical discussion of the findings and the power consumptions are presented in chapter 6.

[9] Complementary metal-oxide-semiconductor (CMOS) is a technology for constructing integrated circuits e.g. microcontrollers, static RAM, microprocessors, RFID tags and other digital logic circuits.

6. Proposal of a Secure RFID Protocol

This section proposes an RFID protocol designed to address the identified security and privacy requirements. Firstly, the design principles are briefly described, following onto the protocol's description including the identification process. Finally, the analysis of the protocol is introduced in order to evaluate the overall achievement against the reviewed prior art in chapter 3, 4. The main emphasis is on the design aspects of the protocol and the security analysis.

6.1 Design principles

The aim is to design an RFID identification protocol for low-cost tags that ensures secure communication between the server and the tag. The server aims to distinguish if the interrogated tag is legitimate or not and thus if valid, obtain the tag's ID without compromising any privacy or security properties.

It is a fact that simple operations such as hash functions and shared secrets are not able to achieve the desired privacy level for passive tags, even if the performance is practical. Authors' throughout the literature review have frequently stated that public-key cryptography (PKC) is essential to overcome the security issues in the emerging technology [56, 65, 66, 67, 71, 85]. However, with this statement the accompanying conclusions state that PKC is expensive for low-cost tags. Thus, authors rarely considered PKC for RFID tags.

This chapter focuses on the design of a new protocol based on PKC that applies ECIES (Elliptic Curve Integrated Encryption Scheme) and ensures security and privacy properties for low-cost RFID tags. The protocol is one-way challenge-response protocol consisting of several parts. The basic privacy model is obtained from Vaudenay's paper [56] that structures the foundation of the new protocol. The model is then extended further to implement other cryptographic primitives such as ECIES, CBC-MAC and AES. The protocol captures narrow-strong and forward privacy, which are the highest possible level of security, and privacy for low-cost RFID tags.

Below are the descriptions of the several parts that construct the design of the new protocol.

ECIES: is a public key encryption scheme used to extend Vaudenay's privacy model [56]. It is semantically secure against chosen-plaintext (CPA) and chosen-ciphertext attacks (CCA) [10, 86, 87]. Recalling the author's statement and proof of security, if the public-key cryptosystem is IND-CCA and IND-CPA secure then the protocol achieves narrow-strong and forward privacy [56].

There are two types of IES schemes, Discrete Logarithmic Integrated Scheme (DLIES) and Elliptic curve Integrated Encryption Scheme (ECIES). This protocol applies ECIES due to the small key size, low memory requirements, and fast computation. Another reason for choosing this scheme is due to its efficiency in tiny hardware implementation. Chapter 5 describes the EC processor that can be applied efficiently for the Elliptic Curve-based public-key cryptosystem i.e. ECIES scheme, used in this protocol [44].

AES: is a cryptographic primitive in particular a symmetric-key encryption that is easily adaptable to hardware requirements. Due to this reason, several RFID protocols utilize condense implementations of AES [46, 47, 84]. AES is characterized as highly secure and well suited for RFID tags. In terms of power consumption and in relation to die size demands, the hardware implementations can be modified accordingly [45].

CBC-MAC: Message Authentication Code (MAC) is a hash function with a secret key. MAC = $h_k(m)$, where *h* is the hash function, *k* is the secret key and *m* is the message. If *m* and *k* are known then computing $h_k(m)$ is achievable. However, if *k* is not provided then computing $h_k(m)$ should be unfeasible. Cipher Block Chaining Message Authentication Code (CBC-MAC) is a method that employs a block cipher in order to construct a message authentication code. A chain of n-bit block message is encrypted with a block cipher algorithm using CBC mode and each encryption output is linked to the previous one. The MAC function is applied onto the last cipher block within the CBC operation mode. CBC-MAC can be used as an alternative to hash functions and modes of operation such as CBC, allow a symmetric block-cipher such as AES. The MAC function allows the verifying party who possess the secret key, to determine if any changes have been made to the plaintext, which consequently ensures data integrity and authenticity. CBC-MACs are largely recognized and commonly used [9].

KDF: A Key Derivation Function is a common way to construct keys for cryptography primitives such as ECIES. KDFs construct secret keys using pseudo-random functions and secret values such as a tag's ID, master keys, passwords etc. KDFs are similar to hash functions except that the output length can be quite large [9, 10].

6.2 Protocol Description

This section introduces the protocol design. The description of the protocol consists of the notation used, the analysis of each part, and the protocol diagram.

The following notation and keys are used for the protocol description:

Notation	Description
T	RFID tag (transponder)
R	RFID reader (transceiver)
R	Random number generated by reader (128 bits)
ID	Identity of tag (128 bits)
K	secret key k (128 bits)
||	Concatenation of two inputs

Keys	Short-term keys	Long-term keys
Public	-	pk
Secret	y, k_1, $k_{2,}$ k_2'	sk, k

6.2.1 Initialisation

The RFID scheme requires some prior steps to setup the system, assign the keys, and store data on the backend server before the protocol can proceed. The initialization phase in this section is similar to the one described in chapter 5.

A scheme needs to firstly generate the long-term public/secret key pair (*pk/sk*) for the system. The server stores *sk* and *pk* is distributed to the tags. Tags do not have access to the *sk* only the reader does. This is an important point for the security analysis of the protocol in later sections.

The basic protocol requires public-key cryptography and this protocol applies ECIES. The key generation of an ECIES scheme executes the following steps [86, 87]:

1. EC domain parameters: (p, a, b, G, n, h) for a curve over prime field or (m, f(x), a, b, G, n, h) curve over binary field. (Note: Chapter 5 describes why binary fields are recommended)

2. Generate random number x ϵ [1, n-1]
3. Secret key: sk 🡐 x
4. Public key: pk 🡐 skG
5. Return pk, sk

Additionally, the system also sets up a unique identification number i.e. ID for each tag that belongs to the system. The length of the ID value has to be large enough so brute force attacks or exhaustive search cannot identify the ID, in this case an ideal length for the ID is 128 bits. The system is also responsible for calling the setup algorithm to generate a secret key *k* for the tag depending on the ID as an input and sets the state of the tag S=(ID, pk, k). The pair (ID, k) is stored in the database. The Setup algorithm can use a KDF to output the specific key *k* from a master key *Km* and the tag's ID using a Pseudo Random Function i.e. k = PRF_{KM}(ID) [56]. The tag stores the specific secret key *k*, the public key *pk* and the unique *ID* which essentially is the state of the tag. The reader stores the secret key *sk* and the (ID, k) pair for each tag [56].

6.2.2 Identification process

The proposed protocol is separates into several sections. Below is the description of how each of these parts that construct the overall design of the protocol.

Part 1: Basic Protocol - the narrow-strong and forward privacy preserving protocol

The diagram demonstrates the process in which the tag and the reader interact in a two-pass, challenge-response protocol and finally the tag outputs the analogous output. (Original diagram [56]).

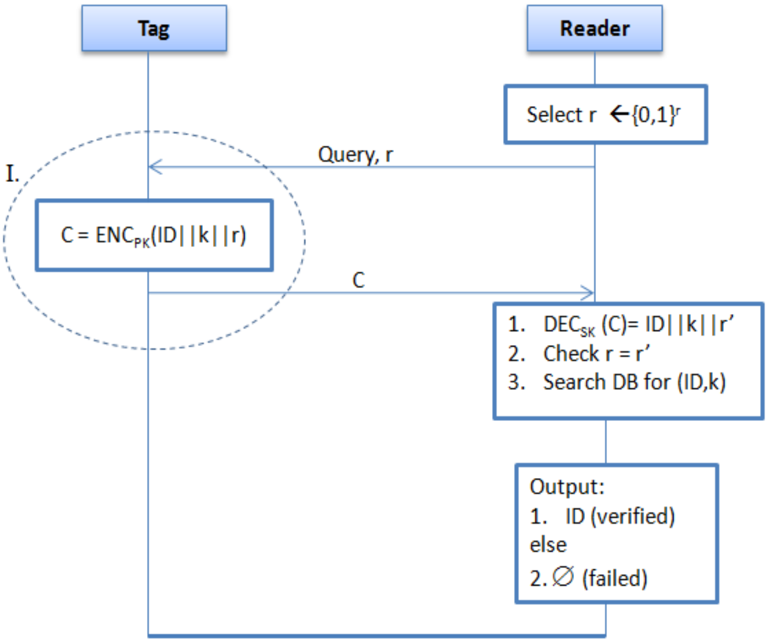


Table 12: Identification diagram

1. The reader generates a random number *r* and forwards it to the tag.
2. The tag will then encrypt the *ID*, the secret key *k* and the random value *r* with the public key *pk* and outputs the ciphertext *C*.
   Note: The public-key encryption at this phase uses ECIES described in the following section.
3. Once the reader receives the ciphertext *C*, it will then decrypt with the secret key *sk* stored on the system and obtain the values *ID*, *k* and *r'*.

4. The reader checks if $r = r'$ is correct.
5. If the value r is correct and the pair (ID, k) is located in the database (DB), the reader outputs the tag’s *ID* → Legitimate tag identified
6. If (ID, $k) \neq (ID, k)$**’** and/or $r \neq r$**’** the reader outputs ∅ which terminates the execution of the protocol’s instance.

Alternative: the KDF generated the secret key k for the tag from a master key k_m and the *ID* using a PRF in the setup phase. The reader can store k_m and execute k = $\mathrm{KDF}_{\mathrm{km}}$(ID) using the *ID* obtained during decryption, and compare the calculated k' with the decrypted k. Since k is derived from the tag’s ID and the system stores the original key k_m used in the KDF, it can verify if the values parsed from the decryption are legitimate by comparing the decrypted values with the stored values, i.e. if $k = k'$. If the tag is valid then the pair (ID, k) is stored in the database. In a real-world implementation, this would reduce the size of the database. For example, the system does not have to initially store all the tag’s ID and corresponding secret keys in the system and then consume a large amount of time locating a match. It can store the master key k_m only and if the values from the KDF algorithm match the decrypted values, it then stores the pair (ID, k) for each tag. This will reduce the performance cost of the back-end server without affecting the level of security [56].

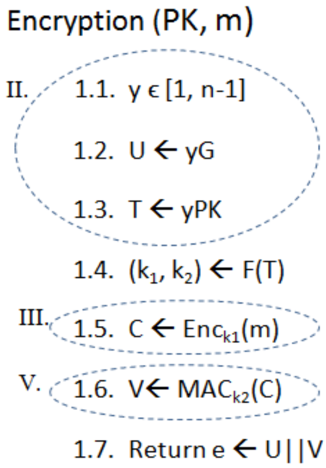
Part 2: Encryption: ECIES scheme (Point I from the identification diagram)

The above diagram illustrates the identification process of the protocol. The design focuses on the identification and authentication of the tag and not the reader since, the two major issues concerning RFID protocol designs are privacy and performance, which both relate to the RFID tags. Hence, the protocol uses Elliptic-Curve-based public-key encryption scheme.

There is optional shared information S_1, S_2, which can be used additionally in this scheme. However, the choice is eliminated. The protocol provides efficient security in order to eliminate these values and even if the computational and storage demands are able to afford them, when design protocols for low-cost RFID tags, even trivial values are considerable.

This scheme allows arbitrary long messages to be encrypted. From the diagram below 1.3, T acts as a key transport mechanism, 1.5 is a symmetric encryption function i.e. AES, 1.6 uses a MAC function that protects against adaptive adversaries [9].

Below is the description of the ECIES encryption which consists of several steps [9, 10, 86, 87].



Encryption (PK, m)

II.
1.1. $y \in [1, n-1]$
1.2. $U \leftarrow yG$
1.3. $T \leftarrow yPK$
1.4. $(k_1, k_2) \leftarrow F(T)$
III.
1.5. $C \leftarrow \mathrm{Enc}_{k1}(m)$
V.
1.6. $V \leftarrow \mathrm{MAC}_{k2}(C)$
1.7. Return $e \leftarrow U||V||C$

1. Initially the scheme generates a random value y such that $y \in [1, n-1]$.
2. The value U and the shared secret value T are calculated.

   Note: step 1.2 and 1.3 are analysed in later on within this chapter.

3. A key derivation function *F* maps group elements to the key space of both the encryption and MAC function [Nigel]. F then computes the key k_1 that will be used for the encryption in step 1.5 and key k_2 used for the MAC in step 1.6.
4. Step 1.5, uses a symmetric encryption, AES that uses k_1 to encrypt the message *m*. The message *m* corresponds to the three values of the tag *ID, K, r* from the above identification diagram in part 1.
5. Step 1.6. applies a MAC function with key k_2, on the output *C* of step 1.5.
6. Finally, the scheme outputs *e* (*e* corresponds to the ciphertext *C* from the identification diagram in part 1) which is the concatenation of the value *U, V* and *C*.
   6.1 the KDF computes keys k_1, k_2 is essentially a hash function that outputs the two keys based on the values U and T e.g. $(k_1,k_2) = H(U||T)$. [9]
   6.2 U is an EC point responsible for the agreement of the ephemeral Diffie-Hellman key T [10].
   6.2 C corresponds to the encryption of the message i.e. *ID, r, k*.
   6.3 V prevents adaptive chosen ciphertext attacks.

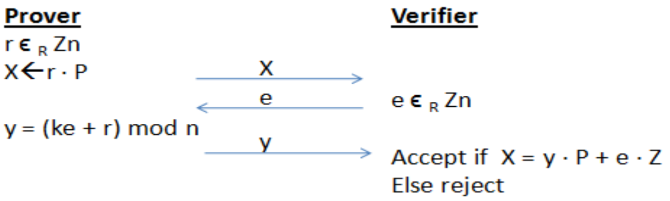
Note: Below is a more detailed description of each point II, III, V from the encryption process.

Part 3: Point II – Calculating the EC points

Recalling for chapter 5, Lee et al. [44], presents an efficient and practical solution for implementing EC processors for RFID protocols. The achieved results of the processor's performance can enhance the calculations of the first part of the ECIES scheme for the proposed protocol. Below is the basic scalar multiplication algorithm, the points are repeatedly doubled and added and the output Q of the algorithm can be used in the protocol to replace the initial values of the ECIES scheme for U←yG and T←yPK. The authors propose an enhanced version of the algorithm in order to integrate with RFID EC-based security protocols analysed towards the end of this chapter.

Important Note: recalling that the public key *pk* from the initialization algorithm was computed from *x* ϵ [1, n-1], *sk = ,* hence *pk = skG*. Essentially *pk* will be computed only once because it is stored in systems permanently i.e. tag's long-term public key. However, the value *T* will be computed for every instance of the protocol but the EC point within the value will only be computed at the setup of the RFID system i.e. T←yPK. This does not reduce the security of the value *T* because a fresh random number is generated for every protocol execution. The value *U* will have to be computed for every identification instance. According to Blake et al. [10], the value U is an EC point and can be compressed to reduce bandwidth.

The algorithm on the left is the basic multiplication algorithm for the Schnorr identification protocol. The value in the Schnorr protocol *r* corresponds to the values *k ϵ [1, n-1]* or *x ϵ [1, n-1]* (from the above description) and the value *X ←rP* corresponds to *U←kG* and *pk←xG*. (Original pseudocode and Schnorr protocol [44])

Basic scalar multiplication-Binary method	Schnorr identification protocol
P: point, k: x-bit integer, $k=(k_{x-1}, k_{x-2},....,k)_{2'}$, k_i ←{0,1} Process: 1. Q = kP; 2. Q←O; 3. For i from t – 1 down to 0 do 4. Q←2Q; 5. If k_i = 1, then Q←Q+P; 6. End for 7. Return Q;	Inputs: q: finite field, a,b: elliptic curve, P: point, n,h: cofactor, k: secret s.t Z=-kP  Prover Verifier r ϵ R Zn X←r · P X e e ϵ R Zn y = (ke + r) mod n y Accept if X = y · P + e · Z Else reject

Part 4: Point III – Advanced Encryption Scheme

The protocol has to execute the following steps in order to complete the symmetric encryption for the ECIES scheme. The proposed method is to use AES with CBC mode. AES-CBC is an authenticated encryption algorithm that can provide authentication and privacy for the encrypted message. The encryption for the ECIES scheme, $C \leftarrow Enc_{k_1}(m)$, requires the message to be encrypted that consists of three parts, *ID, k, r*. The key k_1 generated by the KDF will be used for the following encryption process. Another essential point worth mentioning is the IV[10] for the CBC mode. The value of the IV can be set to 0 and not reduce the security of the protocol since the encryption key is re-generated each time the protocol instantiates a new session.

CBC with IV = 0

$C_1 = Enc_{k1}(ID)$

$C_2 = C_1 \oplus Enc_{k1}(k)$

$C_3 = C_2 \oplus Enc_{k1}(r)$

Output $C = C1||C2||C3 \Leftrightarrow M = ID||k||r$

Part 5: Point V – CBC-MAC

Encryption offers confidentiality of data and assures that an untrusted third party cannot obtain the message that is encrypted. However, it does not ensure data integrity or authenticity. For this reason we use MAC to prevent chosen ciphertext attacks and guarantee that the message transmitted between tag and reader cannot be tampered with. The CBC-MAC for the protocol uses the key k_2 generated from the KDF. It is recommended not to use the same key for the encryption and the MAC function. The key k_2 is used for the encryption and key k_2**'** is used for the MAC. The additional key k_2**'** is applied on the last cipher block i.e. MAC, to provide additional confidentiality of data and to prevent exhaustive key search [9, 10]. The CBC-MAC has to complete the following steps:

$t_1 = Enc_{k2}(C1)$

$t_2 = t_1 \oplus Enc_{k2}(C2)$

$t_3 = t_2 \oplus Enc_{k2}(C3)$

$MAC = Enc_{k2'}(t_3)$

Output $V = Enc_{k2'}(t_3)$

Part 6: Final output of the ECIES scheme and the basic protocol.

The final step of the ECIES will return the value *e* which is the concatenation of the value *U*, the output of the AES encryption *C* and the output *V* of the MAC i.e. $e \leftarrow U||C||V$. This corresponds to the original basic protocol diagram when the tag sends the ciphertext *C* to the reader.

6.3 Analysis of the protocol

This section of the chapter analyses the protocol described above and demonstrates the achieved privacy and security requirements with the corresponding performance of each part of the protocol.

[10] IV: Initialization vector can be described as a block of bits used on the first cipher block on any of the modes of operation (ECB, CBC, OFB and CFB) and ensures a unique encryption for each message without having to re-key. An IV can have variant sizes depending on the cryptographic protocol i.e. size of the encryption key or the size of the block cipher. In order to prevent reduction of security, the IV must not be reused with the same key [9].

6.3.1 Privacy and security

The key advantage of the protocol is the usage of PKC, which achieves the highest level of both security and privacy. Other protocols avoid it due to the additional cost. However, as mentioned in numerous related literatures, in order for an RFID protocol to be secure against the largest possible forms of attacks and offer the desired level of privacy, solutions should be directed towards PKC [56, 65, 66, 67, 85]. With an additional trivial increase in cost, remaining within the cost range of 5-10 cents, the tags can achieve strong privacy and security. The reason lies within the distribution of the keys. In a symmetric protocol or any other protocol of a similar nature, the tag and the reader use shared secret keys and other operations such as hash functions or XOR operations to accomplish secure and successful identification of legitimate tags. If an adversary ascertains the shared key then he can damage the system in several ways. Even if the shared secret key is updated at the end of every session, the protocols are still vulnerable to a desynchronization attack.

In a PKC protocol, the solution is simple. The tag will have the *PK*, which it uses for encryption, and the reader will have *SK* for decryption. The communication channel between reader and the back-end server is assumed secure and the data held in the database, such as the secret key *SK* and any other private data related to tags are secure with some type of access control [56, 85]. The adversary needs the *SK* to decrypt which is impossible to obtain, so the tags privacy and security remains robust. However, strong privacy is impossible [56].

Important points about the proposed protocol are:

1. PKC in an RFID protocol provides the highest level of security [44, 56, 65, 66, 67, 71, 85].
2. An AES-128 encryption is impossible to invert. As a result, the tag's ID, secret key and random nonce are secure even if the communication channel is eavesdropped.
3. Every session uses a fresh random nonce.
4. The keys used for encryption and the MAC function are refreshed for every encryption.
5. MAC function provides data integrity.

Although, security and privacy are two different issues, they are closely related. For example, if the system provides indistinguishability of tags, the privacy requirement of untraceability is thus preserved.

The proposed protocol has the following security and privacy properties.

Tag anonymity: an adversary cannot guess the tag's ID given that the encrypted message send to the reader that will identify the tag consist of the encryption of the *ID*, the secret key *k* and the random value *r*. The encryption scheme used in this protocol is the ECIES and the symmetric encryption used within this scheme is the AES with CBC mode. Consequently an adversary is unable to determine the tag's ID.

User data confidentiality: this property is also referred to as tag information privacy, unauthorized tag reading, privacy and information leakage. The information stored on a tag can contain private information about a user or a company such as medical prescriptions, money etc. An attacker must be prevented from gaining access or altering such information. The tag's information is stored on the system, which is considered secure. The attacker has to first authenticate himself with the server before he actually can access data associated with the tag. However, the identification process of the tag is secure and preserves both information privacy, and protection from tampering with data. The protocol uses a MAC function, $V= MAC_{k2'}(C)$, at the end of the encryption phase, just before it sends *e* (corresponds to *C* in the basic diagram) to the reader. This ensures data integrity i.e. prevents data from being tampered with. The additional key k_2**'** used for the MAC is different from the key k_2 used for the encryption. The additional key prevents exhaustive key search.

Cloning attack: the protocol prevents cloning and tag impersonation attacks. Each tag has its own long-term secret key *k* which is generated by a KDF that takes as an input the tag's ID and produces different secret keys for different tags. For this reason, if one tag is compromised, other tags cannot be affected. Furthermore, an adversary can clone or impersonate a valid tag only if he can access the tag's *ID*, secret key *k*, random value *r* and compute an encryption on all three values. Assuming that the attacker can query a reader and obtain the random value *r* and the public key *PK* for encryption, he then has to obtain a valid ID and a corresponding valid key *k*. This is impossible since he will first have to decrypt at least one ciphertext sent to the reader and in order to decrypt he must know the secret key *SK* which is stored only on the secure server. Additionally, the short-term keys for the encryption process are refreshed for every encryption.

Replay attack: to prevent this type of attack, the adversary must not be able to collect a set of messages and replay then in a valid session with the reader resulting in the validation of an illegitimate tag. This protocol uses a challenge response technique by computing the tag encryptions with the aid of a random value *r* sent from the reader i.e. $tag_{ID}||tag_k||r$. Each tag identification process uses a freshly generated value *r*. The ciphertext *C* includes the freshly generated nonce *r* and so that corresponding *C* cannot be re-used in any other instance. As a result, an attacker cannot replay the messages already used in a session with the reader. Preventing replay attacks consequently prevents spoofing attacks.

Untraceability: a tag is said to be untraceable if an attacker cannot distinguish if he is interacting with the same tag from a previous instance or a different tag. In order to prevent an attacker from tracking a tag, the tag responses must not be identical to any of the previous session. Even though a tag always responds to a reader with the ciphertext C ($\text{Enc}(C)_{SK\ =}$ ID||k||r), an attacker able to eavesdrop on the communication line between reader and tag, cannot observer a relationship between the responses that link to a certain tag. The encryption uses short-term keys and a random nonce *r* that changes for every instance and *C* cannot be linked to any specific tag. In addition, the protocol is IND-CCA and IND-CPA secure since it is based on Vadenay's privacy model [56]. Hence, even if the adversary chooses two tags t_0, t_1 at random and the challenger selects b, and forwards the encryption of C_b (b=1, t_1, b=0, t_0), he cannot distinguish the bit b, i.e. if b=1 or b=0 with probability greater than 0.5% [56]. Also, know as tracking problem, tag location privacy and location tracking attack.

DoS attack: DoS attacks cause the system to be desynchronized when certain transmitted messages are blocked and the system vs. tag update their status inconsistently. This protocol does not divide and send messages at different phases, allowing a jamming signal to block at least one of the messages. This will cause the tag or the reader to update whilst the remainder is unaware about the status update. Such situations lead to desynchronization or any other type of DoS. The protocol is secure against DoS attacks since the communication between the reader and tag consists only of the tag's encrypted message (C = ID||k||r). The tag or the system do not need to update the tag's ID or any other secret information except from the short-term keys which remain internally with each tag.

Forward secrecy: Vaudenay [56], proves that the PKC protocol is IND-CPA secure hence, it provides forward privacy. Even if an adversary eavesdrops on all the communication messages sent between tag and reader, the keys used in the encryption function are short-term keys and they are refreshed for every tag encryption. The adversary cannot obtain any previous messages as each message is independent. The KDF in the ECIES scheme is essentially a one-way hash function that updates the short-term keys k_1 and k_2 as a result, the adversary cannot recover the past secret information or predict future communications. For this reason, backwards privacy is also preserved. Furthermore, recalling that the basic protocol is narrow-strong and forward privacy preserving, Vaudenay's [56], definition of forward privacy is equivalent to backwards privacy. The basic protocol is proven secure against backwards traceability since it is IND-CCA secure. Also, know as forward traceability.

The following comparison table was constructed from the comparison in chapter 4. The table below demonstrates that a public-key cryptosystem is an efficient way to ensure the most security and privacy properties.

Properties vs. Protocols	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	Proposed Protocol
Forward Secrecy	✓	X*	X	P	P (Δ)	✓	✓	X (Δ)	✓ (Δ)	P	X	✓	✓
Cloning Prevention	X	✓	✓	P	✓	✓	✓	X (Δ)	X	✓	✓	X	✓
DoS Prevention	✓	✓	✓	X	X	X	✓	X (Δ)	X	✓	X	X	✓
Replay Attack Prevention	✓	✓	X	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Tag Anonymity	✓	X	X	✓	✓	✓	X	✓	X (Δ)	✓	✓	✓	✓
Untraceability	X	X	X	X	X (Δ)	X (Δ)	X	X	X (Δ)	X	X	X	✓
User Data Confidentiality	✓	X	X	✓	✓	✓	X	✓	X	✓	✓	✓	✓

✓: prevents attack
X: cannot prevent attack
Δ: controversial in literature review
P: partially prevent such attacks (under certain circumstances)
*: P2 consists of 6 protocols, 4 and 6 do not have this property

6.3.2 Performance Analysis

RFID protocols have to be secure, practical and efficient. The table below illustrates an example of a passive tag specifications within the cost range of 5-10cents. Security enhanced tag are slightly more expensive than simple identification tags[11] [45]. The table below is used only as an example, since RFID tag specifications change according to different manufactures and are modernized frequently. (Original figure [71]).

Example an EPC low-cost RFID tag specifications	
Storage:	128-512 bits of read-only storage
Memory:	32-128 bits of volatile read-write memory
Gate Count:	7.5 – 15 K
Security Gate Count Budget:	2.5 – 5 K
Operating Frequency:	13.56 – 900MHz
Scanning Range:	Approx. 1.5 – 6m
Performance:	100 read operations per second
Clock Cycles per Read:	10K clock cycles
Tag Power Source:	Passive – powered by Reader
Power Consumption:	20 µA

In general, the proposed protocol emphasises on the design of the protocol and the achieved security and privacy requirements. The focus is to secure tags efficiently since, readers have the ability to enclose a high-level of security, memory and handle much of the computational workload. The most common requirement readers and servers must attain is scalability. The computational workload should not increase linearly as the number of tags increase in the system. However, this is not an issue in PKC protocols [65-67].The reader is responsible for locating the tag's content once the identification process is successfully completed. In order for this process to be carried out, the

[11] According to Feldhofer[45], in 2006 the ideal estimate price that achieves economic profit for a security-enhanced tag is slightly higher around US$0.10 and for simple identification tags lower than US$0.10.

decryption of the tag's message must first be completed and once the server identifies the tag, it then performs an exhaustive search to locate the corresponding information. The response time for both tags and readers are also limited. According to Li et al. [70] they propose a protocol that is directed towards supply chains dealing with large amounts of tag turnover and servers are forced to locate tag information in a limited time. The authors propose merge-sort and quick-sort algorithms for locating tag information promptly in the back-end server.

There is an obvious trade-off between cost and security when considering the design of RFID protocols. Each proposed protocol has to accentuate on one of the two aspects. If a high level of security is the main priority of the protocol, then the cost will slightly increase and vice-versa. This proposed protocol can be summarized as a slightly more expensive solution for low-cost RFID tags. Nevertheless, as mentioned earlier in the report, many researchers and authors claim that in order to achieve the highest level of security and privacy in an RFID system, a PKC privacy protocol is necessary [44, 56, 65, 66, 67, 71, 85].

Essentially, the tag requires the encryption of the three stored values. The protocol uses ECIES that includes AES encryption and CBC-MAC. This protocol initiates with a reader generating a random nonce from a pseudo-random number generator that protects against replay attacks. The MAC function is used for data integrity and prevention of chosen ciphertext attacks. The different parts of the protocol require different storage and computational requirements.

The initial setup of the system assigns to each tag three values that must be stored i.e. the unique *ID*, the specific secret key *k* and the *pk* used for encryption. A tag needs to store these values and preserver extra additional memory space for computation and communication. According to Lee et al. [44] tags can store the data by hardwiring. This method is characterized as a simple and efficient memory structure, allowing the architectural design of the tag to remain uncomplicated and requires no memory access. The disadvantage of hardwired data is that the architectural design when set as an ASIC[12] cannot be altered and the authors recommended that a part of the data is better to be stored in non-volatile memory [44]. However, in this design, hardwiring the data is a practical way to store the *pk* and the secret key *k* for each tag.

Short-term keys do not have to be stored permanently on the tag i.e. *k1, k2, k2'* and are refreshed for every identification instance. The server stores the secret key *sk*, which is the same for all tags. The reader also stores the pair (ID, k), the specific secret key *k*, the tag's ID and k_m, which is the master key used for the KDF. (l: bit-length of the values i.e. l = 128. m: number of tags.)

Storage of DB (bits)	**Storage of tags** (bits)
2l + 2l *m	3l

Table 13: tag and reader memory requirements

Note: depending on the choice of the systems setup, the server can store the master key k_m and the secret key *sk* and only at the end of every successful identification instance store the pair *(ID,k)*. Alternative, the system can store the secret key *sk*, k_m and the *(ID, k)* pair for all the tags in the system. The former method reduces the size of the database [56].

Key Derivation Function

The KDF is essentially a hash function that produces keys. The hash function that can be used in this situation for example is the SHA-1. According to [74], the processing speed is 1.73 Mbytes per second. Hence, the total amount of time to produce the keys for this protocol is trivial.

[12] Application-Specific Integrated Circuit (ASIC) is an integrated circuit (IC) designed for a specific usage and not for general purpose.

Compact architecture for EC-based security processors for RFID protocols

The proposed protocol is based on an encryption scheme that requires arithmetic of EC values. Recalling from earlier in this chapter, ECIES requires the calculation of two points U and T. Lee et al. [44] findings are ideal for applying to this protocol.

In order to calculate the two values for the ECIES encryption, the memory requirements and access has to be considered due to the limited resources of a passive RFID tag. Table 14, illustrates the total memory need to complete the Schnorr protocol whereas table 15 illustrates the total number of memory access. Table 14 is necessary since it provides an approximation for the memory requirements that associates with the power consumption demand. It is clear from table 15 that version 1 reads from ROM nearly 4 times more than version 2 and 3. The reason is that the version 1 excludes the register for the EC base point and has to reload for every iteration in the Montgomery algorithm consequently the ECP has to load 163 times for a 163-bit key [44].

Each scalar values is 21 bytes and the block size is also 21 bytes. The blocks simplify the processing of the control system and the program due to efficient data management. Calculating the intermediate values memory requirements is also essential. The intermediate values for both the modular multiplication and modular additions have to be temporary stored. For a complete analysis of the storage requirements for these values during the iterations of the algorithms see [44]. However, an important point is that the storage amount for these values is not crucial and tags can afford it.

Below are three tables that present the amount of memory, the memory access, and the overall performance for one EC scalar multiplication. The proposed protocol requires the calculation of *U* and *T* in the ECIES encryption. As mentioned in this chapter, only value *U* will have to be calculated for every session of the protocol. Value *T* is the product of the *pk*, which is set at the system setup phase when the key generation computes the *sk* and *pk*. (key generation x←[n, n-1], *pk*←xG, *sk* ←x then for the encryption *y* ←[n, n-1], *U*←yG, *T*←ypk)

ROM and RAM memory requirements (Original table [44])

Memory	Size
ROM for program	25 bytes
ROM for data	84 bytes (4 blocks * 21 bytes)
RAM	107 bytes (5 blocks + 2 bytes = 5*21 + 2)

Table 14: total memory requirements

* One block of memory = 21 bytes, 2 extra bytes in RAM for modular operations

Schnorr protocol memory access (Original table [44])

	Memory	Read	Write
Version 1	ROM RAM	4.330 3.911	- 3.448
Version 2	ROM RAM	928 3.911	- 3.448
Version 3	ROM RAM	928 3.911	- 3.448

Table 15: Total number of memory access

Performance results (Original table [44])

Below is the table that illustrates the performance results for version 1. All three versions are practical to implement for this protocol. However, version 1 is recommended even though it reads ROM more, it requires the least amount of gate area which when weighing the two factors, memory access or gate area, the later is more crucial for RFID tags. As mentioned by the authors, the time and cycles are to complete the Schnorr protocol and the gate area includes the micro controller, the bus manager and ECP [44].

PKC	Digit Size	Area (gates)	Cycles	CMOS (µm)	Freq. (KHz)	Performance (msec)	Power (µW)	Operations
Type 1 ECC GF(2^{163})	1	12,506	275,816	0.13	1,130	244,08	36,63	Point Multiplications
	2	14,064	144,842		590	245,49	21,55	General
	3	14,729	101,183		411	246,19	15,75	Modular
	4	15,356	78,544		323	243,17	12,08	Operation

AES encryption

The tag encrypts the *ID*, *k* and *r* with the stored *pk*. Given that three values require encryption, this scheme applies CBC mode with AES encryption. AES decryption is not included in the tag due to die-size limitation and mainly it is the server's responsibility to decrypt the ciphertext. The proposed protocol has to perform the AES encryption of the *ID, k* and *r*, and for the CBC-MAC. The MAC function is considered a keyed hash function and XOR functions are simple operations. Both are considered negligible when taking into account the number of cycles i.e. cost [68].

According to Feldhofer et al. [45], an example of an ideal 8-bit architecture AES implementation has a chip area of 3,595 gates and has a consumption of 8.15µA at a frequency of 100kHz. This author takes into consideration the ISO/IEC 18000 standard. The encryption of 128 bits requires about 1000 cycles. Noting that the protocol does not need to replicate the AES circuit it only requires to call the encryption *n* times as a subroutine. It is recommended that the architectural design of the circuit, implements the AES encryption module only once. Again, there exists a trade-off between processing time and die-size, and although both are critical factors, the later can be considered as slightly more crucial. Given that, the processing time has the opportunity to adjust to requirements with an efficient design, whereas, the die-size is decreasing throughout the evolution of RFID tags. (Original tables [45])

AES Encryption	Clock cycles	@100kHz	GE (Gates)
S-Box	280	0.67	395
MixColumns	288	0.41	252
AddRoundKey	144	0.53	90
KeySchedule	304	0.92	161
RAM		4.64	2,337
Controller		0.98	360
Total	1,0016 (~992)	8.15µA	3,628 (~3,595)

Table 16: Components of an AES module

AES-128 Encryption	Clock cycles	@100kHz	GE (Gates)
Feldhofer [45]	992	8.15 µA	3,628
Mangard [46]	64	47.24 µA	10,799
Verbauwhede [47]	10	307 µA	173K

Table 17: Comparison with other AES implementations

6.3.3 Summary

This section presented a new RFID protocol, emphasizing on the design and the security analysis. Observing the comparison outcome of several other RFID protocols with respect to both security and privacy properties, it is noticeable that in order to achieve the highest level of security and consequently ensure privacy, PKC is a possible solution. Scalability is another key property for an RFID system that several previously proposed protocols fail to achieve or disregard it. This design provides the scalability property, due to the public-key approach. Even thought the storage and computational requirements of the protocol requires an additional cost, it can remain within the boundaries of a low-cost RFID tag if the appropriate components are used for the design. This type of protocol guarantees a secure communication channel between the tag and the reader.

7. Conclusion

This chapter presents a brief description of the achieved objectives from each chapter. Towards the end of this chapter, a 'future work' section is presented based on the critical evaluation of the choices and achievements made throughout this dissertation.

This project mainly focuses on the security and privacy issues that form barriers restricting the full adoption of the RFID technology. RFID authentication/identification protocols that utilize various techniques including cryptographic primitives, other simpler functions, and operations were analysed and evaluated. Combining the fundamentals of the technology, the security and privacy issues, and the evaluation of previous protocols, were all necessary topics, which shaped the foundation in order to design and present a secure RFID protocol.

7.1 Critical evaluation and main achievements

Chapter 2 introduced the fundamentals of the technology, including the description of the components found within an RFID system, examples of current applications, operational frequencies, standards and descriptions of the most common cryptographic primitives applied in RFID protocols. Towards the end of this chapter, privacy and security threats were identified and discussed. As previously mentioned, security threats aim at the RFID system whereas privacy issues are formed by the system. Thus, a protocol must ensure the prevention of denial of services attacks, replay attacks, traceability, cloning and spoofing attacks. Resisting these attacks guarantees forward privacy, user data confidentiality and tag anonymity. Meanwhile, it is essential to maintain system scalability and conform to the desired performance requirements for passive tags.

Chapter 3 analyzed and presented twelve previously proposed RFID protocols. The designs and techniques of these protocols vary, and were chosen with the intention of gaining an overall wide comprehension of what, when and how these methods are utilized. The general assumptions can be summarized as follows. Firstly, most protocols use XOR operations, shared secrets such as passwords, keys or hash functions and the authentication process consists of an average of three exchanged messages between reader and tag. Secondly, some authors prefer to divide the workload equally between the reader and tag whereas others load the back-end server. Thirdly, some protocols use either shared common secrets, or updated independent secrets. All of the above approaches bear advantages and disadvantages.

Chapter 4 firstly evaluated the privacy and security properties separately for each of the twelve protocols including a note on the performance and finally compared them with the requirements identified in Chapter 2. The comparison demonstrated whether these protocols provide the identified privacy and security properties. The evaluation revealed that regular techniques such as common shared secrets could lead to cloning attacks or tracking, i.e. tag impersonation or tag compromising attacks that affect other tags sharing the same secret. However, even if tag secrets are independent and updated, scalability, and desynchronization attacks (DoS) start to appear. Other simpler protocols that apply only XOR operations or hash functions, have been frequently proven as weak solutions even though they offer ideal computational performances for passive tags. Understanding how these attacks work and under what circumstances they are mountable on different protocol that use various techniques, results in the prevention of design pitfalls for the proposed protocol in Chapter 6.

Chapter 5 firstly introduced Vaudenay's privacy model [56], then analyzed an efficient example of an EC processor [44] and an example of an AES encryption [45] implementation for passive tags. The analysis of these two examples, prove that schemes such as ECIES, which require both, are feasible for securing low-cost tags since suitable designs exist. During the literature review, other protocols appeared such as [64, 65, 71, 74, 85] that are based either on privacy models or security proofs presented by other authors for the endorsement of their protocol's design. Vaudenay's privacy model was analysed in order to present a hierarchy of privacy models, different adversaries, oracles, proof of security, claimed privacy for passive tags and other essential information relevant to RFID protocols. The narrow-strong and forward privacy-preserving model forms the basic structure for the proposed protocol.

Chapter 6 proposed the design of a new RFID identification protocol based on PKC. The encryption scheme applied in this design is ECIES that has not yet been proposed according to the investigation carried out on related work. The last section of this chapter compares the protocol against the prior art from chapter 4, with respect to both its privacy and security properties. The performance of the protocol is also described, taking into account the discussions from chapter 5. A significant level of privacy abides within the design of the protocol, as it is based on Vaudenay's privacy model [56]. The synopsis drawn from the evaluation, illustrates that the protocol acquires significant advantages since the design can maintain a high level of security, sufficient privacy, and scalability.

The topic of securing low-cost RFID tags requires simplicity and minimalism due to the low computation power, limited die-size and restricted memory. As a result protocols sometimes use similar structures and techniques but with different order and/or additional elements. For instance, the designs of RFID protocols are usually based on the improvements of previous works. Even though the schemes used for the proposed protocol, if considered independently are common i.e. ECIES, AES, privacy model etc, the overall design construction has not been previously presented.

The highlighted outcomes underlined in this area of research can be summarized as follows. There exists an obvious trade-off between security and cost and a need to achieve a desired level of security and privacy for RFID tags. Therefore, the project concluded with the proposal of an RFID protocol. The final chapter mainly focuses on the design, the security, and privacy properties. This protocol does not remain within the usage of simple techniques and operations and therefore the performance analysis compared to previous protocols seems costly. However, even though PKC is avoided for passive tags when considering performance, according to [44, 56, 65, 66, 67, 71, 85], asymmetric-key methods present the highest level of security. Thus, it is worth considering equivalent PKC schemes that are potentially efficient for low-cost tags.

Finally, according to Rotter [17], "Even a secure system will fail if users assume it lacks sufficient security and privacy protection". Hence, the effort of establishing secure and privacy-preserving solution for low-cost RFID tags should be equally balanced with the effort of establishing user trust and awareness.

7.2 Future work

There are many possible areas of research within the field of RFID security and privacy. Some of these topics are as follows:

- This project analysed and evaluated only a certain amount of protocols. A suggestion that can improve the project would be to investigate a wider number of protocols resulting in a more significant evaluation.
- When protocols are firstly presented, their evaluation is usually theoretical. Implementing the protocol in a real-life application that uses RFID, would surely identify the advantages and disadvantages of the design.
- Investigating security for RFID systems and designing protocols, are fields that have recently been accentuated. For this reason, formal models, attack models and security proofs that enhance the analysis of RFID protocols and prove their level of security, are topics under

current research [21, 23, 41, 56, 77, 93].The evaluation of the security and privacy properties for the proposed protocol in chapter 5, are also presented theoretically. Nevertheless, the protocol extends the privacy model of Vadenay [56], which provides security proofs.

- There are several attacks on RFID protocols, which have not yet been considered. Therefore, further investigation on potential attacks of these protocols would be beneficial.
- Furthermore, instead of focusing mainly on the security and privacy properties, a more in-depth analysis of the performance could be considered as a further study. In addition, the architectural design of low-cost tags could also be examined.
- This project focused on general RFID protocols. Various applications, i.e. patient monitoring, libraries, supply chain management etc. necessitate different requirements. As mentioned earlier, not every application requires the same level of security and privacy. Thus, as a further study, focusing on a specific application with specific requirements and aiming to design a protocol that meets those specifications would certainly be valuable.

8. References

[1] B. Glover, and, H. Bhatt, *RFID Essentials.* Sebastopol, California ; Farnham : O'Reilly, 2006.

[2] K. Finkenzeller, *RFID handbook : radio-frequency identification fundamentals and applications.* Chichester : John Wiley, 1999.

[3] G. Wolfram, B. Gampl, and P. Gabriel, *The RFID Roadmap: The Next Steps for Europe.* Berlin ; London : Springer, 2008.

[4] H. Lehpamer, *RFID design principles*. London : Artech House, 2008.

[5] S. Ahson, and M. Ilyas. Ed., *RFID handbook : applications, technology, security, and privacy.* Boca Raton Florida: Taylor & Francis Group, 2008.

[6] V. D. Hunt, A. Puglia, M. Puglia, *RFID : a guide to radio frequency identification*. Hoboken, New Jersey: John Wiley & Sons, Inc. ,2007.

[7] D. Henrici, *RFID Security and Privacy: Concepts, Protocols, and Architectures,* Lecture Notes Electrical Engineering Volume 17. Verlag Berlin Heidelberg Germany: Springer, 2008.

[8] L. Yan, Ed., *The Internet of things: from RFID to the next-generation pervasive networked systems.* New York ; London : Auerbach, 2008.

[9] N. Smart, *Cryptography: An Introduction,* Berkshire: McGraw-Hill Education, 2003.

[10] I. F. Blake, G. Seroussi, and N. P. Smart, Ed., *Advances in Elliptic Curve Cryptography,* London Mathematics Society Lecture Notes Series 317, Cambridge: Cambridge University Press, 2005.

[11] S. L. Garfinkel, A. Juels, and R.Pappu, “RFID Privacy: An Overview of Problems and Proposed Solutions,” *IEEE Security & Privacy*, vol. 3, no.3, pp. 34-43, June 2005.

[12] E. P. Kelly and G. S. Erickson, “RFID tags: commercial applications v. privacy rights,” *Industrial Management & Data Systems*, vol. 105, no.6, pp. 703-713, June 2005.

[13] V. Lockton and R. S. Rosenberg, “RFID: The Next Serious Threat to Privacy,” *Ethics and Information Technology,* vol. 7, no.4, pp. 221-231, December 2005.

[14] A. Juels, “RFID security and privacy: A research survey.” *IEEE Journal on Selected Areas in Communications,* vol.24, no. 2, pp.381–394, February 2006.

[15] G. S. Erickson and E. P. Kelly, “International Aspects of Radio Frequency Identification Tags: Different Approaches to Bridging the Technology/Privacy Divide,” *Knowledge Technology & Policy*, vol. 20, no.2, pp.107-114, August 2007.

[16] R. Goel, “Managing RFID Consumer Privacy and Implementation Barriers*,” Information Systems Security*, vol.16, no.4, pp. 217-223, July 2007.

[17] P. Rotter, “A Framework for Assessing RFID System Security and Privacy Risks,” *IEEE Pervasive Computing*, vol. 7, no.2, pp. 70-77, June 2008.

[18] M. Langheinrich, “A survey of RFID privacy approaches,” *Personal and Ubiquitous Computing*, vol. 13, pp. 413-421, October 2008.

[19] D. Vienhald and A. Wong, “The future of Radio Frequency Identification,” *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 2, no.2, pp. 74-81, August 2007.

[20] M. Ward, R.v. Kranenburg, and G. Backhouse, “RFID: Frequency, standards, adoption and innovation,” *JISC Technology and Standards Watch*, May 2006.[Online]. Available: http://www.jisc.ac.uk/uploaded_documents/TSW0602.pdf. [Accessed: April 22, 2010].

[21] D. R. Thomson, N. Chaudhry, and C. W. Thompson, “RFID security threat model,” *Conference on Applied Research in Information Technology*, March 2006. [Online]. Available: http://csce.uark.edu/~drt/publications/rfid-threats-alar-060303.pdf. [Accessed: April 22, 2010].

[22] A. Juels, R. L. Rivest, and M. Szydlo, “The blocker tag: selective blocking of RFID tags for consumer privacy,” *In Proceedings of the 10th ACM conference on Computer and communication security*, pp. 103-111. October 2003.

[23] T. van Deursen and S. Radomirovic, “Attacks on RFID Protocols”. *Cryptology ePrint Archive*: *Report 2008/310*, July 2008. [Online]. Available: http://eprint.iacr.org/2008/310. [Accessed: April 23, 2010].

[24] H. Y. Chien and C. W. Huang, “A lightweight RFID protocol using substring.” *Lecture Notes in Computer Science: Embedded and Ubiquitous Computing (EUC),* vol. 4808, pp. 422–431, November 2007.

[25] I. J. Kim, E. Y. Choi, and D. H. Lee, “Secure Mobile RFID System against Privacy and Security Problems,” *Security, Privacy and Trust in Pervasive and Ubiquitous Computing, Third International Workshop on Security,* pp.67 – 72, July 2007.

[26] K. Osaka, T. Takagi, K. Yamazaki, and O.Takahashi, “An efficient and secure RFID security method with ownership transfer,” *Lecture Notes in Computer Science: Computational Intelligence and Security*, vol. 4456, pp 778–787, September 2007.

[27] D. Thompson, “RFID technical tutorial,” The Journal of Computing Sciences in Colleges, vol. 21, no. 5, pp. 8–9, 2006. Available: http://www.docstoc.com/docs/2134670/RFID-Technical-Tutorial/. [Accessed: April 20, 2010].

[28] A. M. Wicks, J. K. Visich and S. Li. “Radio Frequency Identification Applications in Hospital Environments” *Hospital Topics*, vol. 84, no. 3, pp. 3-8, Summer 2006.

[29] J. A. Fisher and, T. Monahan, “Tracking the social dimensions of RFID systems in hospitals” *International Journal of Medical Information*, vol. 77, pp. 176 – 183, August 2007.

[30] K. Zetter, “E-Passport Hacker Designs RFID Security Tool,” Wired.com, Apr. 14, 2008. [Online]. Available: http://www.wired.com/threatlevel/2008/04/e-passport-hack/. [Accessed: Apr. 10, 2010].

[31] M. Sparkes, “Hackers clone Elvis’s passport,” *Dennis Publishing Limited,* Oct. 1, 2008. [Online]. Available: http://www.pcpro.co.uk/news/227754/hackers-clone-elviss-passport. [Accessed: Apr. 10, 2010].

[32] M. Sparkes, “Oyster hackers roam London for free,” *Dennis Publishing Limited*, Jan. 23, 2008. [Online]. Available: http://www.pcpro.co.uk/news/207966/oyster-hackers-roam-london-for-free [Accessed: Apr. 10, 2010].

[33] RFID Journal, “RFID Journal Announces Winners of Its 4th Annual Awards*,” RFID Journal LLC*, April 2010. [Online]. Available: http://www.rfidjournal.com/article/view/7543. [Accessed: Apr. 15, 2010].

[34] TI-RFID eNews, “China’s Jiangsu Longton Jail Upgrades Security with TI-RFid,” TI-RFid A TEXAS INSTRUMENT TECHNOLOGY, November 2004. [Online]. Available: http://www.ti.com/rfid/docs/news/eNews/enewsvol25.htm. [Accessed: Apr.15, 2010].

[35] C. Swedbeg, “Macau Casinos Use RFID to Authenticates Chips,” *RFID Journal LLC*, December 2006. [Online]. Available: http://www.rfidjournal.com/article/articleview/2878/1/1/. [Accessed: Apr.18, 2010].

[36] C. Swedberg, “In the U.K., Libraries Switch to Self-Serve,” *RFID Journal LLC,* Feb. 11,2010. [Online]. Available: http://www.rfidjournal.com/article/view/7391/. [Accessed: Apr. 18, 2010].

[37] RFID Journal, “RFID Business Applications,” *RFID Journal LLC*, May 2010. [Online]. Available: http://www.rfidjournal.com/article/articleprint/1334/-1/1. [Accessed: Apr. 19, 2010].

[38] J. Burnell, “New RFID Medical Cabinets Deployed at 50 Hospitals, ” *RFID Journal LLC,* Sep. 18, 2007. [Online]. Available: http://www.rfidupdate.com/articles/index.php?id=1447. [Accessed: Apr. 20, 2010].

[39] M. C. O’Connor, “In Haiti, RFID Brings Relief,” *RFID Journal LLC,* Apr. 8, 2010. [Online]. Available: http://www.rfidjournal.com/article/view/7523/. [Accessed: Apr. 20, 2010].

[40] S. Morton, “Barcelona clubbers get chipped” Sep. 29, 2004. *BBC News/Technology*, [Online]. Available: http://news.bbc.co.uk/1/hi/technology/3697940.stm. [Accessed: Apr. 22, 2010].

[41] A.Juels and S. Weis, “Defining Strong Privacy,” *ACM Transactions on Information and System Security*, vol. 13, No. 1, pp. 1-23, October 2009.

[42] S. Karthikeyan and M. Nesterenko, “RFID Security without Extensive Cryptography,” *SASN '05: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks,* pp. 63-67, November 2005.

[43] H. Daou, A. Kayssi and A. Chehab, “RFID Security Protocols,” *Innovation in Information Technology,2008.* pp. 593-597, December 2008.

[44] Y. K. Lee, K. Sakiyama, L. Batina, I. Verbauwhede, "Elliptic-Curve-Based Security Processor for RFID," *IEEE Transactions on Computers,* pp. 1514-1527, November, 2008.

[45] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, “Strong authentication for RFID systems using the AES Algorithm,” 2004, pp. 357–370. [Online]. Available: http://www.springerlink.com/content/26tmfjfcju58upb2. [Accessed: Aug. 05, 2010].

[46] Mangard, S.; Aigner, M.; Dominikus, S.; , "A highly regular and scalable AES hardware architecture," *Computers, IEEE Transactions on* , vol.52, no.4, pp. 483- 491, April 2003.

[47] I. Verbauwhede, P. Schaumont, and H. Kuo, "Design and performance testing of a 2.29-GB/s Rijndael processor," *Solid-State Circuits, IEEE Journal of* , vol.38, no.3, pp. 569- 572, March 2003.

[48] Q. Yao et al., “Randomizing RFID private authentication,” *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, pp.1-10, March 2009.

[49] L.Tong-Lee, L. Tieyan and L. Yingjiu, “A Security and Performance Evaluation of Hash-Based RFID Protocols,” *Information Security and Cryptology: Lecture Notes in Computer Science,* vol. 5487, pp. 406-424, 2009.

[50] X. Xia; S. Han, "A privacy protection protocol for RFID-enabled supply chain system," *Service Systems and Service Management, 2009. ICSSSM '09. 6th International Conference on* ,pp.305-308, 8-10 June 2009.

[51] S. Cai et al., “Enabling Secure Secret Updating for Unidirectional Key Distribution in RFID-Enabled Supply Chains,” *Information and Communications Security: Lecture Notes in Computer Science*, vol. 5927, pp. 150-164, 2009.

[52] A. Omer, J. Thomas and L. Zhu, “Mutual authentication protocols for RFID systems,” *International Journal of Automation and Computing,* vol. 5, no. 4, pp. 348-365, October 2007.

[53] D. Han et al., “New Security Problem in RFID Systems “Tag Killing”” *Computational Science and its Applications - ICCSA, Lecture Notes in Computer Science,* vol. 3982, pp. 375-384, October 2006.

[54]Sangjin Kim; Jihwan Lim; Heekuck Oh; , "A Hybrid-Based RFID Authentication Protocol Supporting Distributed Database," *Convergence Information Technology, 2007. International Conference on* , vol., no., pp.205-211, November 2007.

[55] Md. E. Hoque et al., "Enhancing Privacy and Security of RFID System with Serverless Authentication and Search Protocols in Pervasive Environments," *Wireless Personal Communications*, vol. 55, no.1, pp.65-79, July 2009.

[56] S. Vaudenay, "On privacy models for RFID." *Advances in Cryptology–ASIACRYPT*, vol, 4833, pp. 68–87, November 2007.

[57] "Efficient Hash-Chain based RFID Privacy Protection Scheme," *In International Conference on Ubiquitous Computing – Ubicomp, Workshop Privacy: Current Status and Future Directions,* 2004.

[58] A. Juels, "Strengthening EPC Tags Against Cloning, " *In WiSe '05: Proceedings of the 4th ACM workshop on Wireless security*, pp. 67-76, March 2005.

[59]BarcodesInc, "Photo of Symbol MC9060-G RFID," *BarcodeInc*, Aug. 2010. [Online]. Available: http://www.barcodesinc.com/symbol/mc9000rfid-zoom.htm. [Accessed: June 10, 2010].

[60] iTech News Net, "Motorola FX7400 series RFID reader," *iTech News Net Latest Gadget News and Reviews,* Nov. 2009. [Online]. Available: http://www.itechnews.net/2009/04/29/motorola-fx7400-series-rfid-reader/. [Accessed: June 10, 2010].

[61] *DPROGRAM.NET, "RFID Chip Conspiracy: Why TV Abandoned Analog,"* Conspiracy Planet*, Jan. 21, 2010*. [Online]. Available:http://www.conspiracyplanet.com/channel.cfm?channelid=74&contentid=6610. [Accessed: June 10, 2010].

[62] J. Edwards, "VeriChip Buys Steel Vault, Creating Micro-Implant Health Record/Credit Score Empire," JustGetThere.us, *Nov.* 12, 2009. [Online]. Available: *http://justgetthere.us/blog/plugin/tag/nick+rockefeller*. [Accessed: June 10, 2010].

[63] Utah Street Networks, Inc, "FDA approves implanted RFID chip for humans," *People Tribet.net*, Feb 16, 2007. [Online]. Available: http://people.tribe.net/348a0adc-d983-40b0-a873-816674073a11/blog/fd36ef55-23ee-40b3-8af1-f4322379d098. [Accessed: June 10, 2010].

[64] M. Jantscher and P. Cole, **"**Security and Authentication Primer**,"** August 2006. [Online]. Available: http://www.autoidlabs.org/single-view/dir/article/6/242/page.html. [Accessed: June 15, 2010].

[65] Y. K. Lee, L. Batina, and I. Verbauwhede, "EC-RAC (ECDLP Based Randomized Access Control): Provably Secure RFID authentication protocol." *IEEE International Conference on RFID 2008*, pp. 97–104, May 2008.

[66] Y. K. Lee, L. Batina, and I. Verbauwhede, "Untraceable RFID Authentication Protocols: Revision of EC-RAC." *IEEE International Conference on RFID 2009*, pp. 178–185, 2009

[67] Y. K. Lee, B. L. Batina, D. Singelee, and I. Verbauwhede, "Low-Cost Untraceable Authentication Protocols for RFID." *Conference On Wireless Network Security: Proceedings of the third ACM conference on Wireless network security*, vol. 3 pp. 55-64, March 2010.

[68] J. Ha, S. J. Moon, J. M. Gonzlez Nieto, and C. Boyd, "Low-cost and strong- security RFID authentication protocol." *Lectures Notes in Computer Services: Emerging Directions in Embedded and Ubiquitous Computing*, vol. 4809, pp. 795–807, November 2007.

[69] Y. K. Lee and I. Verbauwhede, "Secure and Low-cost RFID Authentication Protocols." *2nd IEEE International Workshop on Adaptive Wireless Networks (AwiN),* November 2005.

[70] Y. Li and X. Ding, "Protecting RFID communications in supply chains," In *Proceedings of the 2nd ACM Symposium on information, Computer and Communications Security*, pp. 234-241, March 2007.

[71] A. Juels and S.Weis, "Authenticating pervasive devices with human protocols". *Advances in Cryptology: Lecture Notes in Computer Science*, vol. 3621, pp. 293–308. August 2005.

[72] Y. Seo, H. Lee and K. Kim, “A scalable and untraceable authentication protocol for RFID,” *Emerging Directions in Embedded and Ubiquitous Computing, Lecture Notes in Computer Science,* vol. 4097, pp. 252-261, 2006.

[73] J. Kang, D. Nyang, “RFID Authentication Protocol with Strong Resistance Against Traceability and Denial of Service Attacks,” *Security and Privacy in Ad-hoc and Sensor Networks: Lecture Notes in Computer Science*, vol. 3813, pp. 164-175, 2005.

[74] L. Li et al., “Dynamic Key-Updating: Privacy-Preserving Authentication for RFID Systems,” *Pervasive Computing and Communications, 2007. PerCom '07. Fifth Annual IEEE International Conference on* , pp.13-22, March 2007.

[75] T. Dimitriou, “A lightweight RFID protocol to protect against traceability and cloning attacks.” *SECURECOMM’05: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pp. 59-66. September 2005.

[76] A. Juels, “Minimalist Cryptography for Low-Cost RFID Tags (Extended Abstract)” *Security in Communication Networks*, vol.3352, pp. 149-164, 2005.

[77] G. Avoine, “Adversarial Model for Radio Frequency Identification,” Cryptology ePrint Archive, Report 2005/049, July 2008. [Online]. Available: http://eprint.iacr.org/2005/049.pdf [Accessed: July 03, 2010].

[78] H. Gilbert, H. Sibert, and M. Robshaw, “An active attack against HB+: a provably secure lightweight authentication protocol,” *IEEE Electronic Letters*, vol. 41, no.21, pp. 1-2, October 2005.

[79] J. Fan, J. Hermans and F. Vercauteren, “On the claimed privacy of EC-RAC III,” *Cryptology ePrint Archive*: *Report 2010/132*, March 2010. [Online]. Available: http://eprint.iacr.org/2010/132. [Accessed: July 10, 2010].

[80] Z. Golebiewski, K. Majcher, F. Zagorski, M. Zawada, “Practical Attacks on HB and HB+ Protocols,” *Cryptology ePrint Archive: Report 2008/241*, May 2008. [Online]. Available: http://eprint.iacr.org/2008/241. [Accessed: April 24, 2010].

[81] T. V. Deursen and S. Radomirovic, “Untraceable RFID Protocols are not trivially composable: Attacks on the revision of EC-RAC,” *Cryptology ePrint Archive*: *Report 2009/332*, July 2009. [Online]. Available: http://eprint.iacr.org/2009/332.pdf [Accessed: July 15, 2010].

[82] I. Damgard and M. O. Pedersen, “RFID Security: Tradeoffs between Security and Efficiency,” *Cryptology ePrint Archive: Report 2006/234*, July 2006. [Online]. Available: http://eprint.iacr.org/2006/234.pdf [Accessed: July 13, 2010].

[83] T. V. Deursen, and S. Radomirovic, “Security of RFID Protocols – A Case Study,” *Electronic Notes in Theoretical Computer Science*, vol. 244, pp. 41-52, July 2009.

[84] B. Toiruul and K. Lee, “An Advanced Mutual-Authentication Algorithm Using AES for RFID Systems,” *International Journal of Computer Science and Network Secuiry,* vol. 6, no. 9, September 2009.

[85] M. Ohkubo, K. Suzuki and S. Kinoshita, “Cryptographic Approach to “Privacy-Friendly” Tags,” in *RFID Privacy Workshop*, MIT, 2003.

[86] N. P. Smart, “The Exact Security of ECIES in the Generic Group Model,” *Cryptography and Coding: Lecture Notes in Computer Science,* vol. 2260, pp.73-84, 2001

[87] Certicom Research “SEC 1: Elliptic Curve Cryptography,” *Standards for Efficient Cryptography*, September 2010. [Online]. Available: www.secg.org/collateral/sec1_final.pdf [Accessed: August 05, 2010].

[88] J. Bringer, H. Chabanne, and T. Icart, “Cryptanalysis of EC-RAC, a RFID identification protocol,” *Cryptology and Network Security: Lecture Notes in Computer Science*, vol. 5339, pp.149-161, 2008.

[89] S. A. Weis, et al., *“*Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems*,*” *Lecture Notes in Computer Science,* vol. 2802, pp. 201-212, 2004.

[90] X. Gao, et al., “An Approach to Security and Privacy of RFID System for Supply Chain,” *E-Commerce Technology for Dynamic E-Business, 2004. IEEE International Conference on* , pp.164-168, September 2004.

[91] M. Feldhofer, “An Authentication Protocol in a Security Layer for RFID Smart Tags, ” *Electrotechnical Conference, 2004. MELECON 2004. Proceedings of the 12th IEEE Mediterranean* , vol.2, pp. 759-762, May 2004.

[92] N. Hopper and M. Blum, “Secure Human Identification Protocols,” *Adv. in Cryptology — Asiacrypt 2001, LNCS*, vol. 2248, pp. 52–66, 2001.

[93] G. Lowe, “A hierarchy of authentication specifications,” *Computer Security Foundations Workshop, 1997. Proceedings., 10th* , pp. 31-43, August 2002.

[94] P. Peris-Lopez, et al., “LMAP: A Real Lightweight Mutual Authentication Protocol for Low-cost RFID tags,” In: *Proc. of 2nd Workshop on RFID Security*, July 2006.