SIMULATION AND VERIFICATION OF SECURITY ATTACKS ON LIGHT-WEIGHT RFID PROTOCOLS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SAMAN AHMED

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTERS OF SCIENCE
IN
CRYPTOGRAPHY

AUGUST 2014

Approval of the thesis:

SIMULATION AND VERIFICATION OF SECURITY ATTACKS ON LIGHT-WEIGHT RFID PROTOCOLS

submitted by **SAMAN AHMED** in partial fulfillment of the requirements for the degree of **Masters of Science in Department of Cryptography**, **Middle East Technical University** by,

Prof. Dr. Bülent Karasözen _____
Director, Graduate School of Applied Mathematics

Prof. Dr. Ferruh Őzbudak _____
Head of Department, Cryptography

Assoc.Prof. Dr. Melek DikerYücel _____
Supervisor, Graduate School of Applied Mathematics

Examining Committee Members:

Prof. Dr. Ferruh Őzbudak (Head of the Examining Committee) _____
Head of Department, Cryptography

Assoc.Prof. Dr. Melek DikerYücel (Supervisor) _____
Graduate School of Applied Mathematics, METU

Prof. Dr. Ersan Akyıldız _____
Graduate School of Applied Mathematics, METU

Assoc.Prof.Dr Ali Doğanaksoy _____
Graduate School of Applied Mathematics, METU

Dr.Çağdaş Çalık _____
Graduate School of Applied Mathematics, METU

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : SAMAN AHMED

Signature :

ABSTRACT

SIMULATION AND VERIFICATION OF SECURITY ATTACKS ON LIGHT-WEIGHT RFID PROTOCOLS

Ahmed, Saman
M.Sc., Department of Cryptography
Supervisor : Assoc.Prof. Dr. Melek DikerYücel

August, 2014, 63 pages

Radio Frequency Identification (RFID) technology is fast reaching all avenues of application. From retail to warehousing, tracking farm animals to monitoring medicine dosage in human body, traffic control to airport baggage control, it is penetrating all forums and industries and providing ease of deployment and automated visibility and management of inventories which was not possible with traditional barcodes.

Along with its superiority over barcodes, RFID systems are also required to be cost-effective to be fully integrated in commercial products. This means that the price of single tag has to be brought down enough so that it becomes feasible for large scale utilization. Consequently, the commercial tag, which is usually priced between 0.05-0.1 USD, can only contain basic hardware for few logical functions. Hence instead of complex cryptographic solutions, light-weight protocols that require relatively small amount of computations need to be designed to provide security.

In this thesis, ultra-lightweight RFID security protocols are examined in light of their security provisions and susceptibility to certain attacks. In particular, LMAP^{++} protocol presented by Tieyan Li in 2008, is studied and a full disclosure attack presented by Wang Shao-hui et al. in 2012 is applied. It has been seen that this attack is successful and certain new observations have been highlighted.

Next, Strong Authentication and Strong Integrity (SASI) protocol, proposed by Hung-Yu Chien in 2007, and Gossamer protocol, proposed by Pedro Peris-Lopez et al. in 2009 to overcome the weakness of LMAP^{++} have been studied; and a probabilistic attack presented by Eslam Gamal Ahmed in 2010 has been applied on Gossamer. It has been found in this thesis that this attack is unsuccessful, and Gossamer Protocol is in fact, secure against this attack. Further, a Denial-of-Service attack has also been proposed against Gossamer protocol.

Keywords: RFID, Security, Protocol, LMAP, Gossamer

ÖZ

BENZETİMİ VE ÜZERİNE GÜVENLİK ATAKLARIN DOĞRULAMASI
IŞIK-AĞIRLIK RFID PROTOCOLS

Ahmed, Saman
Yüksek Lisans, Kriptografi Bölümü
Danışman: Doç. Dr. Melek Diker Yücel

Ağustos, 2014, 63 Sayfa

Radyo Frekansı ile Tanımlama (RFID) teknolojisi, pek çok uygulama alanına hızla yayılmaktadır. Perakendeden depolamaya, çiftlik hayvanı takibinden insan vücuduna uygun ilaç dozajının gözlemlenmesine, trafik kontrolünden havaalanı bagaj kontrolüne kadar her alana giren RFID sayesinde, geçmiş barkod sistemlerinde mümkün olmayan ölçüde dağıtım kolaylığı, stok yönetimi ve takip edilebilirlik sağlanmıştır.

RFID sistemlerinin barkodlardaki kullanım üstünlüğe rağmen, ticari ürünlere tamamen entegre edilebilmeleri için maliyetin uygun olması gereklidir. Büyük ölçekli kullanımların anlam kazanması için kart fiyatının olabildiğince düşük tutulmalıdır. Genellikle 0,05$ - 0,1$ arasında fiyatlandırılan ticari kartlar sadece temel birkaç mantıksal fonksiyon donanımını içerebilir. Bu nedenle, güvenliği sağlayan karmaşık kriptografik çözümler yerine, görece az hesaplama içeren hafif-ağırlıklı protokoller tasarlanmalıdır.

Bu tezde, ultra-hafif RFID güvenlik protokolleri, sağladıkları güvenlik koşulları ve belirli saldırılara karşı duyarlılıkları ışığında incelenmiştir. Özel olarak, 2008 yılında Tieyan Li tarafından tasarlanan LMAP^{++} protokolü üzerinde çalışılmış ve 2012'de Wang Shao-hui ile arkadaşlarının LMAP^{++} için önerdiği tam-açığa-çıkarma-saldırısı uygulanmıştır. Çalışmamız sonucunda, bu saldırının başarılı olduğu görülerek bazı yeni gözlemler vurgulanmıştır.

Daha sonra, LMAP^{++}'ın zayıflıklarını gidermek için 2007 yılında Hung-Yu Chien tarafından tasarlanan Güçlü Kimlik Doğrulama ve Güçlü Bütünlük (SASI) protokolü ve 2009'da Pedro Peris-Lopez ve arkadaşları tarafından önerilen Gossamer protokolü üzerinde çalışılmıştır. 2010 yılında Eslam Gamal Ahmed tarafından sunulan olasılık atağı Gossamer protokolüne uygulanarak saldırının başarısız olduğu, Gossamer protokolünün olasılık atağına karşı dayanıklılığı gösterilmiştir. Ayrıca, Gossamer protokolüne karşı bir Hizmet-Reddi (Denial-of-Service) saldırısı önerilmiştir.

Anahtar Kelimeler: RFID, Güvenlik, Protokol, LMAP, Gossamer

ACKNOWLEDGMENTS

I would like to extend my gratitude to my supervisor, Assoc. Prof. Dr. Melek D. Yücel for her guidance, insight, patience, support and a little constructive critisism, without which, I could not have completed my thesis.

I also appreciate the efficient staff, research assistants and students of Institute of Applied Mathematics, particularly, Nejla Erdoğdu, Çağdaş Çalık, Ahmet Sınak and Rusydi Hasan for extending all possible administrative and technical support.

Last but not the least, I want to thank my parents, husband and close friends, who looked after my twin babies so that I could persue and complete my studies.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

In the world where Radio Frequency Identification (RFID) is becoming ubiquitous, it is becoming increasingly difficult for a novice user to even understand all the risks he is being placed under unknowingly, what to talk of guarding his privacy against these risks. The thoroughly informed attacker, on the other hand, has complete tools at his disposal to play havoc in the lives of RFID product consumers. Even the manufacturers of RFID products and the inventory owners of RFID tagged items have the technical ability to monitor their products even after they are no longer in their ownership.

In order to promote fair practices in the utilization of RFID devices, Garfinkel proposed an 'RFID Bill of Rights' [1], to put forward the rights of RFID system consumers and purchasers. His proposal included the following points:

- **The Right of Knowledge:** A person should be made aware if a product contains RFID tag.
- **The Right of Removal:** A person should be able to de-activate, destroy or remove the associated RFID tags, upon purchase of an item.
- **The Right of Alternatives:** Upon declining an RFID-enabled device or choosing to de-activate it, a person should not be penalized through a restriction, which he would not be facing if he retained his RFID tags in action. (For example, refusal to return a product to shop, not being able to travel on a road which accepts only RFID toll.)

- **The Right of Amendment:** A person should be able to know the exact contents of his tag's information and he should be able to correct it, if it is incorrect.
- **The Right of Whereabouts:** A person should know why, where and when his tag's information is being transmitted, read or used.

The aim of proving security on the RFID system is to uphold these basic principles and to prevent attackers from using the system components for their malicious activities. The use of cryptographic encryption schemes is ideally suited for this security purpose but the low cost of commercial tag does not allow complex solutions.

The rest of this chapter outlines the key concepts in RFID technology and its security. In Chapter 2, different security proposals presented over the time are reviewed. In Chapter 3, LMAP^{++} is presented [2], and its disclosure attack [3] is verified. In Chapter 4, the Gossamer protocol proposed by P. Peris-Lopez et al[4] and the related attack by E. G. Ahmed [5] are reviewed. We show in Section 4.4 that the attack proposed in [5] does not work. Additionally, we propose a Denial-of-Service (DoS) attack against Gossamer protocol in Section 4.5.

1.1 Automatic Identification and Data Capture

Automatic Identification and Data Capture (AIDC) also commonly referred to as "Automatic Identification," "Auto-ID," and "Automatic Data Capture" is the collection of technologies through which data is sensed from an object automatically and compared with an electronic database to provide identification. These typically include the sensing of images, sound, video or Radio Frequency (RF) emissions from the object, forming the basis of bar codes, RFID, iris and facial recognition system (biometrics), magnetic stripes, Optical Character Recognition (OCR), smart cards and voice recognition systems.

1.2 Barcode

A barcode system consists of an optical scanner which is scrubbed across some machine-readable data printed on an object, and as it scans, the data is interpreted and used for identifying the object.

The original barcodes systems use a system of parallel lines whose width and spacing is varied to represent the data. This is called the one-dimensional (1D) or linear representation (Fig. 1.1). Modern barcode data also consists of two dimensional (2D) systems formed through geometrical shapes such as rectangles, hexagons and dots (Fig 1.2).

Figure 1.1: 1D barcode data.

Figure 1.2:2D barcode data.

1.3 Radio-Frequency Identification (RFID)

RFID is a technology through which electromagnetic waves are used to transfer data between two devices for the specific purposes of identifying and tracking objects containing those devices.

RFID Technology has been around since 1970, but until recently, it has been too expensive to use on a large scale. The original tags had complex systems of metal coils, antennae and glass (Fig. 1.3).However, modern systems are composed of a silicon based microchip, antenna and an optional battery (Fig. 1.4).

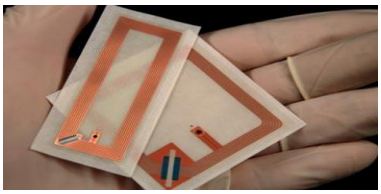


Figure 1.3: Coil and Antenna Tag Assembly.

The ability to provide identification without physical contact creates vast avenues of application, which have previously been exploited through bar codes.

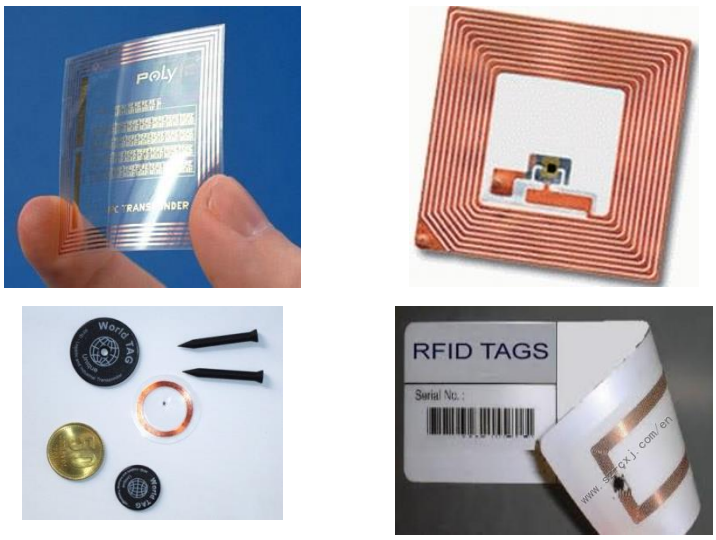


Figure 1.4: Modern RFID Tag.

RFID have several advantages over barcodes, such as follows:-

- Unlike barcodes, in which each line item is required to be manually aligned by a laborer, RFID readers do not work in optical domain and hence, they offer much higher level of automation as they do not require a direct line of sight.
- This also results in robustness, as the electronic reader can be packaged in plastic cover, since it is not required to be exposed on the outside of the object.
- The RFID tag can be re-written electronically, while the information printed on a barcode cannot be altered.
- RFID also provide much higher rates of 40+ tags per second, while the barcode typically scans two items per second, when it is perfectly aligned. Alignment issues further increase the scan time considerably in barcodes.

- RFID can also provide greater distance coverage, upto 300 feet, while barcodes are limited up to 15 feet.
- The security of RFID is superior to barcodes, which are displayed on the out and can be easily reproduced. Encryption and password protection can secure the RFID information stored on a tag, along with a host of added security features added electronically, since there is no physical limitation.

All these features make RFID the most suitable technology for a wide range of applications such as access management, warehousing, tracking humans and animals, toll collection, contactless payment to name a few.

1.4 Real Life Applications of RFID

Some areas, where RFID is being used greatly listed below:

Retail: Provides warehousing and supply chain visibility, resulting in real-time asset management, contactless payment.

Access Management: Limiting entry in secured environments

Tracking of goods, people, livestock, airport baggage, medicines inside human body etc.

Traffic: Toll collection on roads is carried out automatically. Vehicles have an RFID tag mounted on them and as they pass a point, the reader notes the tag's ID and the toll is automatically deducted from the account linked for that vehicle.

Smartdust (for massively distributed sensor networks): Large numbers of small-sized RFID tags are distributed over a large area and can form a network to sense and transmit required information, e.g., the volcanic activity, enemy movement in battlefield, etc.

Some areas where RFID is envisioned to influence vastly are as follows:

- Real time inventory management through the use of a ‘smart’ shelf, which can identify the addition and removal of items.
- Self-aware storage, which knows all its contents, will eradicate the need to look for stored items.
- Lost and found would be simplified where each found item will identify itself and the owner can be located or the owner can query his lost item for its location.
- Automated homes, where the refrigerator signals to the supplier to ship a used-up item, microwave-oven reads cooking instructions from food packages, self-aware medicine cabinets and laundry machines. See [6] and [7] for more examples.
- Over-speeding vehicles are automatically fined when they crosses a speed gun linked to an RFID reader.
- Monitoring medicine dosage of patients.

1.5 Working Mechanism

A typical RFID system consists of the following components [8]:

- RFID tag to carry identification data.
- The reader to read and write tag data.
- The electronic database to store tag records
- Communication channel of RF medium

At a basic level, each tag works in the same way (Fig.1.5), as follows:

Step 1. A reader continuously sends out inquiries when it is active.

Step 2. When a tag comes in vicinity of the reader, the electromagnetic energy from the reader activates the tag through induction circuitry.

Step 3. The tag replies to the reader by sending its data on modulated carrier, using energy harvested from the reader or power harvested from its own battery.

Step 4. The reader receives this signal and demodulates it to interpret the data.

Step 5. The information is then compared to the database to carry out tag identification.

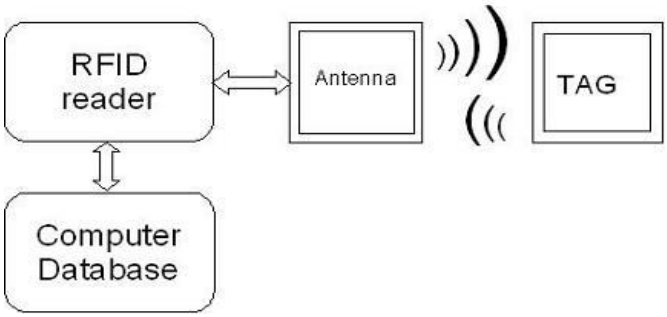


Figure 1.5: RFID Working Mechanism.

1.6 RFID Tags

Tags are physically labeled onto every object, which is required to be identified through the system. They are primarily composed of a microchip, which carries out storage and computations, and is attached to a coupling element, such as coil antenna for transmission and reception. Based upon its utilization and cost, a tag can be designed as read-only, write-once-read-many (WORM), or re-writable.

In terms of onboard system power, tags can be classified as active, semi-active or passive RFID tags [8].

Active Tag: A tag powered by its own battery is an active tag. It has the ability to initiate its own communication, as well as carrying out sense functionality in idle mode.

Passive Tag: A tag which does not contain its own battery and uses the electromagnetic energy of the inquiring reader to induce its own power is a passive tag. The passive tag receives electromagnetic energy from the communication signal of the reader when it interrogates the tag. The tag uses inductive coupling, in the form of a coil or antenna to induce an electric current. This current charges the

onboard capacitor, providing power and voltage to the tag for carrying out its operations. Hence, this type of tag is completely powered-off when there is no reader to provide the energy.

Semi-Active Tag: A semi-active or semi-passive tag contains its own battery but relies on the reader's electromagnetic energy to charge it. This concept is being applied to increase the life of RFID systems in hard to reach environments, where physical access to the tag becomes almost impossible, e.g. Smart-dust.

Passive tags have shortest ranges but are typically the easiest and cheapest to manufacture. Both the cost and range increase with semi-active tags, and the active tags provide the largest ranges with a matching high cost.

1.7 Tag Classification Based Upon Functionality

Tags may fall into classes 0-4, with added functionality in each higher class [8].

Class 0: Class 0 tags are passive devices which do not contain any programmed logic or memory. They only announce their presence without offering any specific identification. This type of tag is commonly found on CDs and library books.

Class 1: Class 1 tags contain read-only memory (ROM) or write-once-read-many memory (WORM). They give specific product-based identification. They are mostly passive, but may be semi-active or active as well.

Class 2: Tags falling in this category have read-write memory, because of which they can log events. Their ability to be written again allows them to be re-used for a new product. They are primarily semi-active or active but can be passive as well.

Class 3: Class 3 tags contain sensors for measuring environmental variables such as temperature, acceleration, motion etc. They are semi-active or active devices and contain writable memory to take readings

Table 1.1: Tag Functionality Classes, reproduced from [8]

Class	Application	Memory	Power Source	Features
0	Anti-Shoplift Tags	None	Passive	Article Surveillance
1	EPC*	Read-Only	Any	Identification Only
2	EPC*	Read-Write	Any	Data Logging
3	Sensor Tags	Read-Write	Semi-Passive or Active	Environmental Sensors
4	Smart Dust	Read-Write	Active	Ad Hoc Networking

* Electronic Product Code (EPC) Tags

Class 4: Tags with networking capabilities to communicate and create ad-hoc network with surrounding tags fall in this category. Since they may have to initiate communication with other devices; they need to be active.

Table 1.1 gives an overview of tag classes with respect to their functionality and applications [8].

1.8 EPC Tags

Electronic Product Code (EPC) is a class of RFID tags which is extremely cheap to produce, for extremely large scale use on each line item in inventory management, supply chain control and retail checkout. It appears that this type will replace barcode's Universal Product Code (UPC) in retail. EPC tags are essentially passive devices with 1000-10,000 gate count, and may use as less as 10μW of

power for each identification operation. They may have 96-512 bit read-only or fixed time re-writable memory [8].

Hence, EPC stags cannot include public key cryptosystems such as RSA and NTRU, encryption schemes like DES and AES or even standard hash functions like SHA-1, due to the high computational requirements of these algorithms.

1.9 RFID Readers

Readers or interrogators query the tag to provide its identification through the RF medium. Since readers are much fewer in number, compared to the tags, it is cost effective to provide the bulk of complex computation at the reader; hence they generally have a much greater processing power than the tags. The readers are also linked to the electronic database to provide authentication and identification of the tag.

Readers also require much greater storage capacity to record the data from large number of tags. Readers may be mobile, hand held device or fixed to a location depending upon the application.

A reader, providing basic functionality may cost only 5 USD. However a portable reader with wireless connection to database can cost between 100-200 USD [8]. The relatively fewer number of readers can justify their higher price.

1.10 Database

The back-end database may provide information about the product, movement history and key management related to a particular tag. For the purpose of designing a protocol, the database and reader are assumed as a single entity with a secure connection existing between them.

1.11 Communication Channel

The frequencies on which the tag/reader communicate are specified by the regulating government agency. When the reader modulates this carrier frequency with information, it creates side-bands. Modulation of the carrier reduces the power available for harvesting at the tag. The spread of the band and the transmission power are restricted by government. Hence the amount of information that can be transmitted from the reader to the tag is limited.

The power received at the tag also decreases with the distance d from the reader at a rate of$1/d^3$or $1/d^2$, depending upon whether it has been harvested from electromagnetic coupling of near field or the far field respectively [8].

1.12 Coding and Modulation

The data which is to be transmitted between the tag and reader, must be encoded in a suitable scheme, the choice of coding scheme and modulation govern how much bandwidth is utilized and how much power is available at the tag. Since the forward channel from the reader to the tag is active all the time, it is subject to government regulations on bandwidth. However, the backward channel, from the tag to the reader only operates when inquired and hence can support a coding scheme giving high bandwidth.

Apart from providing high power and narrow bandwidth on forward channel, another constraint is to provide collision detection on the backward channel where multiple tags may be competing for the same reader. Hence the coding scheme used for forward channel may be Pulse Position Modulation (PPM) (Fig 1.6) due to its low bandwidth and higher power, while the backward channel is favored by Manchester Coding (Fig 1.7) due to its collision detection capability at a higher bandwidth [8].

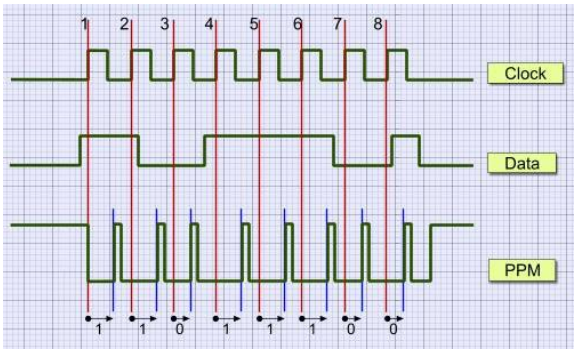


Figure 1.6: Pulse Position Modulation (PPM).

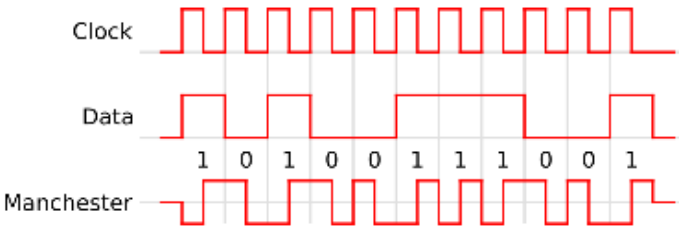


Figure 1.7: Manchester Coding Scheme.

The data code can be modulated through Amplitude Shift Keying (ASK), Frequency Shift Keying (FSK) or Phase Shift Keying (PSK), (Fig 1.8) depending upon design requirements of power consumption, bandwidth and reliability [8].

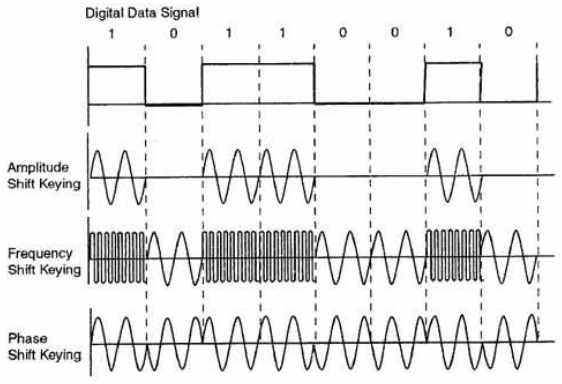


Figure 1.8: Amplitude Shift Keying (ASK), Frequency Shift Keying (FSK) and Phase Shift Keying (PSK).

1.13 Frequency Channel

The Industrial-Scientific-Medical (ISM) bands are utilized for commercial RFID systems. The most common frequencies are:

- 13.56MHz
- 902-928MHz
- 9-135kHz

Each of these bands is subject to different regulations on bandwidth and power. In order to avoid the backward channel from being jammed by the high powered forward channel, it may be transmitted on a different frequency or on a subcarrier [8].

1.14 Collision Detection

There is also a need to provide collision detection and avoidance for the backward channel where a large number of tags may compete for a single reader. Since the tags are not able to communicate among themselves, this task has to be undertaken by the reader. The reader sends out a query and all the tags wait for an amount of time, selected randomly from a fixed duration before replying. Since this waiting time is random, there is very small possibility that any two tags may have selected the same time and reply simultaneously. If this happens, the reader detects interference in the backward channel, and signals all tags to increase the fixed duration or the waiting window. Thus, in a densely populated scenario, the tags have to wait longer on average, and the throughput decreases. The EPC Global standard for RFID specifies Slotted ALOHA anti-collision scheme[9].

The International Organization for Standardization / International Electrotechnical Commission (ISO/IEC) 15693 standard for RFID vicinity cards (range upto 50cm) specifies Amplitude Shift Keying with 10% or 100% modulation index and PPM for the forward channel while ASK with 100% modulation index or FSK and Manchester coding is to be used for backward channel[10]. A

summary of standards and their specifications are presented in Table 1.2.

1.15 Tag Constraints

With so many diverse fields of use, and tags required for each item, it is essential that the cost of a single tag is economic enough to be used commercially. High end application for high-value items utilize tags priced in the range of 0.50–1.00 USD, where the higher price permits the use of strong cryptographic solutions running on high-end hardware to provide tag security. However, a single low-end tag is required to be priced between 0.05–0.10 USD, in order to be used significantly in commercial market [8]. Consider for example, a retail owner who has to buy 500,000,000 tags, for each line item on his inventory. A tag costing only 0.01 USD less will result in him saving 5,000,000 USD. Hence he will have to justify the additional price he has to pay for any extra features like enhanced security.

Consequently, commercial tags are produced with low specifications to meet the price constraint. These limitations include the following:

- Limited storage capacity of few hundred bits
- Limited number of logical functions due to only a few thousand gates present on the chip.
- No on-board battery, hence the tag only powers up while being interrogated by the reader; it cannot carry out background calculations in idle time.
- Low gain on board antenna and power constraints limiting the transmission distance and quality.
- Packaging in paper which does not provide tamper resistance.
- As per rule of thumb, every 1,000 gates on the chip increase the price by 1 cent [8].

Table 1.2: RF Specifications for Different Standards.

Standard		**ISO/IEC14443 [11]**	**ISO/IEC15693 [10]**	**EPCGlobal[9]**
Application		Proximity contactless cards upto10cm	Vicinity contactless cards upto 50cm	For all goods, upto 2m
For-ward chan-nel	**Carrier**	fc=13.56MHz	fc=13.56MHz	860-960MHz (depending on local regulations) divided into channels (200kHz, 500kHz)
	Modula-tion	ASK 100%	ASK 10% or 100%	ASK
	Coding	Modified Miller	PPM ‘1 out of 4’ or ‘1 out of 256’	PIE
	Band-width	106-847kbps	26,48kbps or 1.65kbps	26.7-128kbps
Back-ward chan-nel	**Sub-carrier**	fc/16- fc /128	Single: fc/32 Dual: fc/28,fc/32	
	Modulat ion	Load modulation		ASK or PSK
	Coding	OOK or Manchester	**One subcarrier**: (subcarrier)(unmod ulated time)=0, reverse=1 **Two subcarriers**: (subcarr1)(subcarr2)=0, reverse=1	Reader decides
	Band-width	106-847kbps	Low: 0.6kbps, High: 26kbps	FM0 baseband (40-640kbps) Miller of a subcarrier(5-320kbps)

All these constraints make it infeasible to use strong cryptographic solutions to secure a commercial tag, hence lightweight algorithms and protocols, based upon simple logical operations are designed to function in the constraint hardware.

1.16 Security Risks

With such vast deployment and so many limitations, RFID tags are prime targets for cryptographic attacks. Tags can be hacked to provide access to secure locations. They can provide location information for the tagged objector person, which may be dangerous in wrong hands. Combining details about a tag from multiple tag locations can lead to an individual's profile of movement, transactions and social interactions.

By eavesdropping in on legitimate conversation, an attacker can use the recording to fake it in any setting; e.g., toll collection can be avoided by fooling the system. In devices permitting read/write access, the attacker can manipulate the data contents of the tag. In an environment relying heavily on RFID systems for tracking, Denial of Service (DoS) attacks can disrupt all system operations.

The information obtained from tags in a department store can be used to make inventories and establish supply and demand patterns, which may be useful to rival agencies. It can assist in theft of a product and its replacement with others by forging a fake tag.

Hence any tag/reader system has to be designed to cater for these security concerns and provide adequate safeguard against attacks, while staying in its low-price budget.

1.17 Types of Attacks

Mitrokotsa et al. present a detailed classification of RFID attacks, based upon the layer of operation [12]. A brief summary of attacks important in the context of this work is presented as follows.

- **Eavesdropping**: When an attacker listens to a legitimate conversation, in an attempt to find the secrets.
- **Unauthorized Tag Reading and Tag Cloning**: The attacker is able to impersonate as the tag and communicate with the reader; or when it acts as the reader and queries the tag, in order to obtain its secret. Mutual authentication is used to nullify this attack.
- **Tracking:** The attacker listens to the transmissions over time, in anticipation that by learning past data, he can predict and impersonate the future response.
- **Replay Attack**: The attacker replays the previously eavesdropped messages from a legitimate reader/tag, in order to provide its own authentication.
- **De-synchronization Attack**: The attacker forces the reader/tag to fall onto different states of a secret variable. This happens when it prevents either the tag or the reader from receiving the other's authentication, when its own authentication has already been transmitted and established. Consequently, one device updates the state of the secret variable, while the one not having received the authentication stays behind.
- **Disclosure Attack:** It is an attempt to find the secret information by transmission of data modified from previously eavesdropped messages, to see the impact on the reply.

1.18 Forward Security

This is the ability of the reader/tag system to maintain data integrity in the event of all the previous data on the tag being divulged. In case the tag becomes physically compromised, and the attacker obtains all secret information on it, even then he cannot predict the future messages.

1.19 Hash Function

A Hash Function converts a data of arbitrary length to a fixed size dataset, called the hash value or simply, the hash. The function has the following properties:

- Given a data, it is easy to calculate its hash. Thus, if a message m has a hash value h, such that h=hash(m), then knowing m, it is easy to compute h.
- Given a hash, it is computationally infeasible to calculate the actual data. From the previous analogy, this means that knowing h, it is impractical to calculate m.
- Small changes in data result in large variations in the hash. See Fig 1.9 for the SHA-1 hash function
- It is not feasible to find two different data having the same hash. If h_1=hash(m_1) and h_2=hash(m_2), then if $m_1 \neq m_2$, then practically$h_1 \neq h_2$.

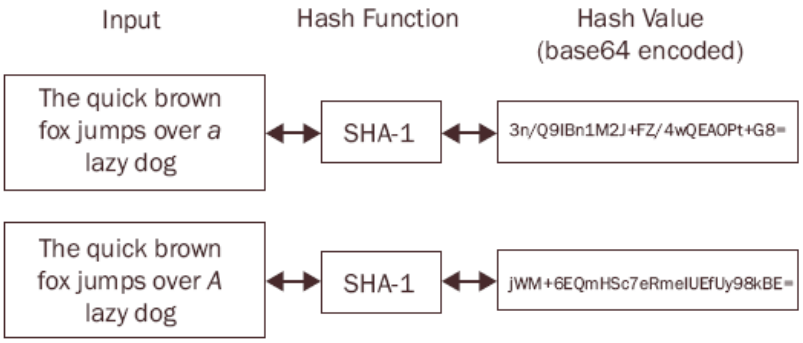


Figure 1.9: SHA-1 Hash Values for Marginally Different Texts.

Hash functions find applications in all areas where information is to be transmitted, keeping its integrity intact. Some areas are the following:

Digital Signatures: Messages are sent along with pre-calculated hash values. Upon reception, the hash values are calculated again and compared to the original ones. If there has been even a slight change

in the message, the hashes will not match and the receiver will detect that the message has been changed after sending.

Password Protection and Verifications: Servers store their users' passwords in the form of hashes, since storing all passwords in original form can lead to a security breach. However, it is not easy to find the original password from the stored hash. When a password is entered, its hash is freshly calculated and compared to the one stored against that user in the database. Upon matching, password is accepted.

Identification: Hashes allow for fast look up of data from an established database, hence they can be used as file or data identifiers.

Key Derivation and Pseudorandom Number Generation (PRNGs): Hashes are used to generate new keys or pseudorandom numbers from an original number or key.

CHAPTER 2

HISTORICAL DEVELOPMENTS

The object of securing low-cost RFID devices is extremely challenging since they are highly resource-restrained and incapable of supporting strong cryptography.

Over the years, a number of lightweight schemes, relying on functions like simple hash, reduced AES, bit-wise operations, such as modular additions, exclusive OR, AND etc. have been proposed to provide security and their subsequent weaknesses have also been identified. Our work summarizes these proposals to provide an understanding of the major concepts in these developments.

This chapter is organized as follows: Sections2.1, 2.2 and 2.3 describe Hash lock schemes [8]and NH algorithm [13,14]. Various other proposals are listed in Section 2.4. Silent Tree Walking algorithm by Weis [8] is reviewed in Section 2.5 while the adaptation of Hopper Blum (HB) algorithm to RFID [15] is discussed in Section 2.6. Section 2.7 presents different approaches for consumer security and Section 2.8 describes blocker tags, both proposed by Juelet al. [16]. The chapter is concluded in Section 2.9 with a description of Ultra-light Mutual Authentication Protocols (UMAP) presented by Peris-Lopez et al. [17,18,19].

2.1 Hash Lock

In [8], security proposals based upon hash function were presented to protect the integrity of the reader/tag channel in low-cost RFID devices.

In this scheme, the tag is 'locked' by the reader, meaning that it does not offer any functionality, until it gets 'unlocked'. The tag's ID and hash(ID)(called meta-ID) are stored in reader database as well as on the tag. The tag stays in 'locked' state in which it replies to all queries with its meta-ID. The legitimate reader compares the received meta-ID in its database to find the corresponding ID. It transmits the ID to the tag which then calculates hash(ID). If hash(ID) matches meta-ID, tag is unlocked.

2.2 Randomized Hash Lock

In this adapted version of hash lock[8], the tag finds a random nonce R and calculates meta-ID as the hash of the concatenation of ID and R. So,

$$\text{meta-ID} = \text{hash} (\text{ID} \,||\, \text{R})$$

The tag transmits the concatenated message, R|| meta-ID in locked state. The legitimate reader calculates hash (ID_k || R) for all k number of tags in its database. When it finds a match, it transmits the corresponding ID back to the tag to unlock it.

This scheme is not suitable if the number of tags associated in the database is very large, since it will take a lot of time to calculate the hash for all tags, as required here. Also, the requirement of both hash function generation and pseudorandom number generation on the tag increases its price. Most importantly, there is no provision for secrecy as the ID may be eavesdropped since it is transmitted openly. The privacy may be improved by transmitting hash(ID) by the reader in the last stage, instead of ID.

Most widely used hash functions include [20] and NSA developed hash functions, SHA0[21] and SHA1[22]. However, attacks has been found against MD5 [23] and SHA0 [24,25,26] and SHA1[27] from 2004 to 2009. A new function, Keccak [28] was selected as SHA2 after winning the National Institute of Standards & Technology (NIST) hash

function competition in 2012. SHA3 is to be proposed, based upon a version of Keccak.

Solutions, based upon hash function have been proposed in [29,30,31].These solutions, however, do not solve the practical problem of running hash functions on the strictly limited capacity of 5000 to 10,000 logic gates present on the tag, of which only 250-3000 can be devoted to overall security tasks. In [13],a lightweight hash function family NH has been proposed and improved in [14]which is described in the following section.

2.3 NH Algorithm

The message is padded to increase its bit-length to$2k\times64$, where k is the smallest possible integer. It is then broken down in blocks of 64 bits. These blocks are fed into Adder_1 and Adder_2 as m_1 and $m_2$consecutively. The algorithm accepts 64 bit keys k_1 and k_2. The algorithm functions as shown in Fig. 2.1. The final Adder_3 accumulates the results of all previous multiplications in a 128 bit result. Instead of a 64 bit scheme, the algorithm may be adapted to any given number of bits.

This algorithm [13] requires minimal hardware to implement but its security is not guaranteed against passive eavesdroppers.

2.4 Other Proposals

Schemes based upon concatenated hash function chains have been proposed [32] by Ohkubo et al., along with the introduction of pseudorandom key called Index-Pseudonyms (IDS). Juels [33] uses **Pseudonym Throttling** in one-time pseudo keys but the number of keys is exhaustive and needs to be renewed. Molnar et al. [34] present tree-based private authentication, which reduces the workload of the reader with n number of tags from $O(n)$ to $O(\log n)$. This construction is further improved by Molnar et al. [35] to include ownership-transfer and time-limited delegation of IDS to reader from a Trusted Center

(TC). AES based mutual authentication protocol is used by Feldhofer et al. [36, 37].

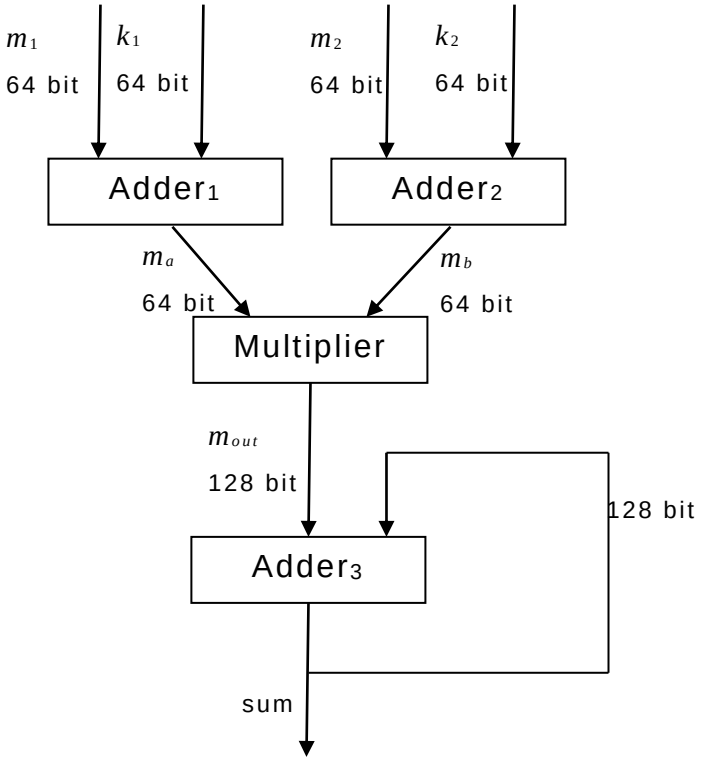


Figure 2.1: Simplified Functional Diagram for NH, reproduced from [13]

2.5 Silent Tree-Walking

Weis [8] identifies that the high power transmissions from the reader are more susceptible to be eavesdropped than the lower power transmission from the passive tag.

He explains a binary anti-collision scheme, in which the reader broadcasts the selected tag's data bit, XORed with the previous bit (Fig. 2.2).

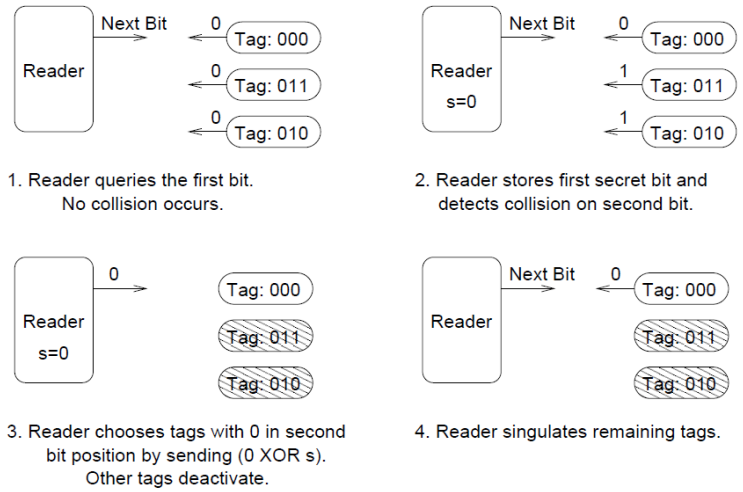


Figure 2.2: Reader Selecting Tag 000 with Tree Walking Algorithm, from[8]

2.6 Hopper-Blum Authentication

The Hopper Blum (HB) [38] uses the concept of human-computer protocol, which can be computed by a human and does not require complex computations. Drawing parallel between the low computational capabilities of a human to that of a low-cost tag, Weis applies HB authentication for RFID tags [15].

The HB protocol is based upon the Learning Parity with Noise problem (LPN), in which two parties, say Alice and Bob who share an n-bit secret a number x. Suppose it is required that Alice wants to authenticate the identity of Bob, in such a way that neither party has to divulge the secret x for security purposes. Alice sends Bob a random n-bit message called a. Both parties calculate the dot product $z = a.x$ and Bob returns his calculated z to Alice, who confirms Bob's identity if his result is same as hers. In this scheme, a listener can only find the correct secret x half the time, since $z \in \{0,1\}$. However, an eavesdropper who captures $O(n)$ rounds can deduce x entirely. To solve this, Bob intentionally sends the wrong bit for k rounds out of total q rounds and Alice accepts result if less than kq parities are wrong.

HB protocol can be very efficient for low-cost devices since it only requires memory to store n-bit x and simple AND and XOR logic to carry out the dot product. The random noise can be generated by diode breakdown, shot, or thermal noise etc.

This problem is close to that of finding the closest vector to a random linear error correcting code [39]. LPN is an NP-hard problem; whose hardness over random instances is not verified. Using the best known algorithm for solving LPN, proposed by Blum, Kalai, and Wasserman (BKW) [40] it has been estimated by Weis [15] that following number of steps will be taken roughly by BKW for solving this problem:

- 2^{56} steps of computation for keys of length $n = 128$
- 2^{64}steps for $n = 160$
- 2^{80} steps for $n = 224$.

2.7 Different Approaches for Consumer Privacy

Juels et al. propose some out-of-the-box solutions to secure tags from unwanted tracking [16]. These include the following:

Kill Tag Approach: The tag is issued command to permanently disable or 'kill' itself before being placed in the hands of consumers. This approach is easy to implement but it may not be the desired universal solution because some products might need to be returned and re-entered in inventory. Various other present or future scenarios for keeping the tags alive are presented in [16].

Faraday Cage: The tag to be concealed is placed in a metal foil which stops certain frequencies of electromagnetic waves from penetration. This approach might be useful for small, valuable items but it may not be practical for large bulky items.

Active Jamming: Tags which are to be shielded should be accompanied by active jammers to block all RFID readers functioning

in the vicinity. It might be illegal if the jamming power is high and may disrupt all legitimate readers and RFID applications under attack.

2.8 Blocker Tags

Blocker tags introduced by Juels [16], work with the tree walking algorithm (described earlier in this chapter) to provide tag security from unwanted readers. Blocker tags carry out passive jamming, by simulating all the 2^k possible RFID tag serial numbers, for serial numbers having k bits. This means that when the tree walking algorithm queries the next bit, the blocker tag transmits both 0 and 1 all the time. If the reader has time and computational capability to complete the algorithm in this circumstance, it would detect all possible 2^ktags to be present, effectively blocking the actual tag. The blocker can be configured to selectively block only a subset of all the tags.

The blocker tag is a powerful tool which may be used for consumer privacy in a positive manner against unwarranted readers. However, same may be used to block legitimate readers and assist in malicious activities like product theft.

2.9 Ultra Lightweight Mutual Authentication Protocols

In 2006, a family of Ultra-lightweight Mutual Authentication Protocols (UMAP) was proposed by Peris-Lopez, Hernandez-Castro, Estevez-Tapiador and Ribagorda. These included the following:

- EMAP: Efficient Mutual Authentication Protocol [17]
- LMAP: Lightweight Mutual Authentication Protocol[18]
- M^2AP:Minimalist Mutual Authentication Protocol[19]

In 2008, Tieyan Li, presented the LMAP^{++} authentication protocol [2], for the low cost RFID tag to avoid the weakness of LMAP. Being one of the main protocols that we examine in this work, LMAP^{++} authentication protocol is described in Chapter 3.

2.9.1 EMAP: Efficient Mutual Authentication Protocol

EMAP [17] introduces the 96 bit pseudorandom key called IDS, which is used in the reader database to identify the tag, instead of transmitting the actual, fixed tag identification (ID) before authentication is carried out. The tag and the reader also share four common 96 bit keys, K_1, K_2, $K_3$and K_4.For any variable X, $X(n)$ denotes the value of X at state n.

The communication is started by reader by saying hello. When the tag receives this inquiry, it replies with its IDS. The reader compares the IDS in its database to find a match. It generates 96 bit random numbers n_1 and n_2 and sends messages A, B and C with the corresponding K_1, K_2 and K_3;where $\oplus$, $\wedge$ and $\vee$, respectively denote bitwise XOR, AND and bitwise OR operations.

$$A = \mathrm{IDS}(n) \oplus K_1(n) \oplus n_1 \quad (2.1)$$

$$B = (\mathrm{IDS}(n) \vee K_2(n) \oplus n_1 \quad (2.2)$$

$$C = \mathrm{IDS}(n) \oplus K_3(n) \oplus n_2 \quad (2.3)$$

The tag extracts the random number $n_1$from$A$ and B. If they match, the tag authenticates the reader. In order to present its own authentication to the reader, the tag sends message D and E, after extracting $n_2$from message C and using K_4 and K_1as follows.

$$D = (\mathrm{IDS}(n) \wedge K_4(n) \oplus n_2 \quad (2.4)$$

$$E = (\mathrm{IDS}(n) \wedge n_1 \vee n_2) \oplus \mathrm{ID} \oplus K_1(n) \oplus K_2(n) \oplus K_3(n) \oplus K_4(n) \quad (2.5)$$

Once both the tag and reader are mutually authenticated, IDS and keys are updated as follows

$$\mathrm{IDS}(n+1) = \mathrm{IDS}(n) \oplus n_2 \oplus K_1(n) \quad (2.6)$$

$$K_1(n+1) = K_1(n) \oplus n_2 \oplus (\mathrm{ID}(1{:}48) || \mathrm{Fp}(K_4(n)) || \mathrm{Fp}(K_3(n))) \quad (2.7)$$

$$K_2(n+1) = K_2(n) \oplus n_2 \oplus (\mathrm{Fp}(K_1(\mathrm{n}) || \mathrm{Fp}(K_4(n)) || \mathrm{ID}(49{:}96)) \quad (2.8)$$

$K_3(n+1) = K_3(n) \oplus n_1 \oplus (\mathrm{ID}(1:48)||\mathrm{Fp}(K_4(n))||\mathrm{Fp}(K_2(n)))$ (2.9)

$K_4(n+1) = K_4(n) \oplus n_1 \oplus (\mathrm{Fp}(K_3(n))||\mathrm{Fp}(K_1(n))||\mathrm{ID}(49:96)),$ (2.10)

where ID(1:48) and ID(49:96) denote the first and last 48 bits of ID respectively.

Parity Function Fp(*X*): The 96-bit number *X* is divided in twenty four 4-bit blocks. A parity bit is taken from each block, getting 24 parity bits. E.g., taking$K_4(n)$= [0001001011001110……11010001],writing the 96 bits of $K_4(n)$in 4 bit blocks, Table 2.1 gives the value of 24 bit parity function as Fp($K_4(n)$)= [1101…10]

Table 2.1: Parity Function

	Block 1	**Block 2**	**Block 3**	**Block 4**	**….**	**Block 23**	**Block 24**
$K_4(n)$=	0001	0010	1100	1110	…..	1101	0101
Parity bit for each block (even)	1	1	0	1	….	1	0
Fp($K_4(n)$)	1	1	0	1	…..	1	0

Hence The EMAP protocol can be implemented in a less than 1000 logic gates, but requires a small read-only memory ROM or small portion or rewritable memory EEPROM for storing the IDS and keys.

2.9.2 LMAP: Lightweight Mutual Authentication Protocol

The LMAP protocol [18] is similar to EMAP with slight variations in the tag's response. Using similar terminology as Section 2.9.1, the protocol uses only three keys K_1, K_2 and K_3 and the tag replies back only message *D* for its authentication, using additions modulo 2^m, for a bit-size *m*.

$D = (\mathrm{IDS}(n) + \mathrm{ID}) \oplus n_1 \oplus n_2$ (2.11)

2.9.3 LMAP^{+}

In a simple extension of LMAP proposed in same paper [18], the reader will store a fixed number of potential IDS associated with a particular tag, in order to prevent de-synchronization attacks. When the tag's reply message *D* is blocked, the reader will store the value of IDS(n+1) in its database as the potential IDS. Each time, when the tag's IDS does not match, it is compared to potential IDS list to carry out authentication steps. After a complete mutual authentication step, the list of potential IDS is reset.

2.9.4 M^{2}AP: Minimalist Mutual Authentication Protocol

The M^{2}AP [19] is also similar to the other two protocols with slight differences in the application of the logical operators.

These three protocols are efficient enough to be implemented in 250-3000 gates and provide better security than hash function based NH algorithms [13] and [14]. However, it was found in [41], [42] and [43], that the secret ID can be disclosed after eavesdropping some rounds of communication. This is based upon the fact that there are only modular additions and logical operations, which do not provide diffusion property; and each bit only affects those bits, which are on the more significant position of that bit; hence the least significant bits are independent.

CHAPTER 3

MODELING OF EFFICIENT FULL DISCLOSURE ATTACK ON LMAP^{++} PROTOCOL

3.1 Introduction

RFIDs are replacing barcodes at every forum and are being used in wide-spread applications requiring tracking and authorization. However, their ubiquitous utilization poses a considerable security and privacy risk to organizations and individuals using them. Since, a typical tag provides ID upon being queried by any reader, it can be easily hacked by an attacker which reads out the data of the tag and copies it to an unauthorized tag. This can result in vulnerabilities to eavesdropping, location privacy, spoofing, or denial of service.

The low-cost, low memory and low computational capabilities of these tags prevent expensive cryptographic solutions to secure the stored information from unauthorized users. Hence light-weight authentication protocols are used to prevent unauthorized readers from gaining access as well as providing mutual authentication for authentic readers and authentic tags. As mentioned in Chapter 2, extremely lightweight protocols have been proposed in 2006 by Peris-Lopez, Hernandez-Castro, Estevez-Tapiador, Ribagorda [17,18,19] and in 2007 by T. Li and G. Wang [44], focusing on bitwise operations like modular additions, XOR, AND etc., compatible with the limited rewritable memory and limited computation power of passive RFID tags. In 2008, Tieyan Li, presented the LMAP^{++} authentication protocol [2] for the low cost RFID tag to avoid the weakness of LMAP.

On the other hand, passive attacks on light-weight protocols take advantage of their reliance on simple bitwise operations. In 2012, W. Shao-hui et al. [3] described the algorithm to attack LMAP^{++}, based upon eavesdropped information. Described attack falls in the network-transport layer, as shown in Fig. 3.1.

In this work, the attack on LMAP^{++} presented in [3] is verified and found that minimum 5 rounds are required for guessing the secret random number, instead of 20, as originally considered in [3].

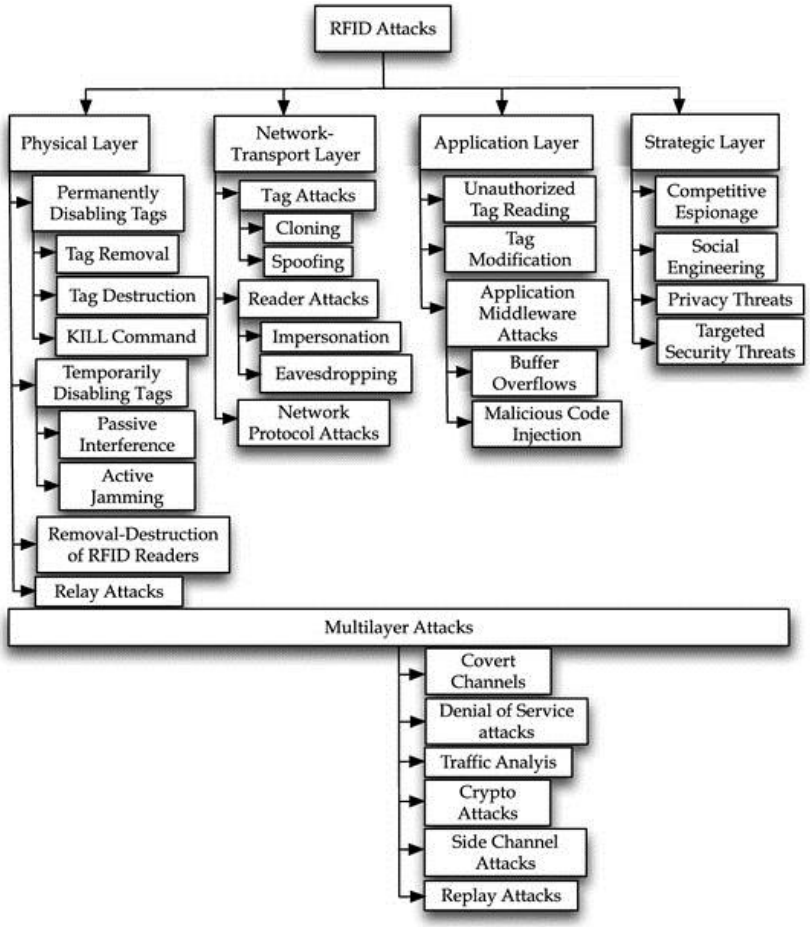


Figure 3.1: Classification of RFID Attacks, Based on Layer of Operation reproduced from [12].

This chapter is organized as follows. In Section 3.2, LMAP^{++} protocol [2] is overviewed. The algorithm of the attack [3] mounted on LMAP^{++} is described in Section 3.3. Simulations and experimental

results are elaborated in Section 3.4 and 3.5. The chapter is concluded in Section 3.6.

3.2 LMAP^{++} Protocol

In the LMAP^{++} protocol, each tag has a static identifier ID. In addition, each i^{th} tag has a pseudonym IDS(i),which is used in transmissions, to avoid giving out the real tag ID, and two secret keys; i.e., $K_1(i)$ and $K_2(i)$. All three of these get updated after each successful run of the protocol. The values of $IDS_{tag(i)}$, $K_{1tag(i)}$, and $K_{2tag(i)}$ at the n'th successful run of protocol are denoted by $IDS^{(n)}_{tag(i)}$, $K_1{}^{(n)}_{tag(i)}$, and $K_2{}^{(n)}_{tag(i)}$ respectively. Hence, in this protocol, the tag and the reader save the values $ID_{tag(i)}$, $IDS^{(n)}_{tag(i)}$, $K_1{}^{(n)}_{tag(i)}$, $K_2{}^{(n)}_{tag(i)}$ in a table T_T. The information exchange between the reader and the tag is as described below.

3.2.1 Tag Identification

Reader sends hello to the tag. The tag replies by sending its $IDS^{(n)}_{tag(i)}$. On receiving a $IDS^{(n)}_{tag(i)}$ from a tag, the reader looks into T_T . If $IDS^{(n)}_{tag(i)}$ is present in T_T, the reader extracts the related $K_1{}^{(n)}_{tag(i)}$ and $K_2{}^{(n)}_{tag(i)}$. Otherwise, the reader terminates the session.

3.2.2 Mutual Authentication

A random number r is first generated by the reader. With $r$and the keys K_1 and K_2, the reader then generates the messages A and B, using equations (3.1) and (3.2), and sends them to the tag; where $\oplus$ denotes XOR and + denotes addition mod 2^k (k is the bit size).

$$A = (IDS \oplus K_1) + r \quad (3.1)$$

$$B = IDS + K_2 \oplus r \quad (3.2)$$

The tag, upon receiving the messages A and B, calculates r_1 from A and r_2 from B, using the secret keys K_1 and K_2 separately. If r_1 equals to r_2, reader is authenticated.

The tag now sends C to the reader, using its static ID. The tag is successfully authenticated, if the reader can find a valid ID from message C.

$$C = (\text{IDS} + (\text{ID} \oplus r)) \oplus (K_1 + K_2 + r) \quad (3.3)$$

3.2.3 Key Updating

After mutual authentication, both the tag and reader will use the following equations to update the IDS, K_1 and K_2 :

$$\text{IDS}^{(new)} = ((\text{IDS} + K_1) \oplus r) + ((\text{ID} + K_2) \oplus r) \quad (3.4)$$

$$K_1^{(new)} = (K_1 \oplus r) + (\text{IDS} + K_1 + \text{ID}) \quad (3.5)$$

$$K_2^{(new)} = (K_2 \oplus r) + (\text{IDS} + K_1 + \text{ID}) \quad (3.6)$$

3.3 The Attack Algorithm for LMAP^{++}

The attack takes advantage of the inherent weakness of bit-wise operations (such as $\oplus$: bitwise XOR, $\wedge$: bitwise AND, $\vee$: bitwise OR, and + modulo 2^kaddition). Each bit will affect the same bit position or the bit with higher index positions, but have no influence on its lower bit, so the least significant bits are independent.

The data can be broken into chunks, and exhaustive search is run on the least significant chunk, independently. The result of the lowest chunk can be concatenated with the more significant chunk to run exhaustive search on that size. For example, for an n bit data, divided into 4 chunks, the exhaustive search is reduced from $2^n$to $4 \times 2^{n/4}$.

Prior to the attack, the attacker assumes a fixed number of conversations (m) between the RFID tag and the authentic reader, which will be spoofed by the unauthentic reader. Hence the reader has knowledge of following variables:

- IDS(1) till IDS(m)
- A(1) till A(m)
- B(1) till B(m)
- C(1) till C(m)

The attack proceeds as follows:

(a) The unauthentic reader will make a random guess for a fixed k number of least significant bits of the variable $r^{(1)}$, say $r^{(1)}(1{:}k)$. Using this guess it will calculate $K_1^{(1)}(1{:}k), K_2^{(1)}(1{:}k), \mathrm{ID}(1{:}k)$, using equations (3.1), (3.2) and (3.3).

(b) It will further calculate the new updated $\mathrm{IDS}^{(2)}(1{:}k)$ and keys $K_1^{(2)}(1{:}k), K_2^{(2)}(1{:}k)$, using equations (3.4) (3.5) and (3.6), and find the new $r^{(2)}(1{:}k)$, by substituting the eavesdropped value of $A^{(2)}(1{:}k)$ and calculated values of $\mathrm{IDS}^{(2)}(1{:}k)$ and $K_1^{(2)}(1{:}k)$ in equation (3.1).

(c) Finally, the unauthentic reader will calculate $B^{(2)}(1{:}k), C^{(2)}(1{:}k)$ and $\mathrm{IDS}^{(3)}(1{:}k)$, and verify them against the eavesdropped values.

If they do not match, the guess is omitted and the above steps (a), (b) and (c) are repeated with a new guess. In this way, the calculations further proceed until all 2^k guesses of $r^{(1)}(1{:}k)$, are exhausted. For each guess, the attacker matches till $B^{(m)}(1{:}k), C^{(m)}(1{:}k)$, and $\mathrm{IDS}^{(m)}(1{:}k)$, noting only those guesses of $r^{(1)}(1{:}k)$, which provide perfect match for all variables.

After guessing $r^{(1)}(1{:}k)$, in this way, the attacker now guesses $r^{(1)}(k{+}1{:}2k)$, and uses the concatenation of $r^{(1)}(1{:}k)||\mathrm{r}^{(1)}(k{+}1{:}2k)$ to essentially follow the same steps as before to find $r^{(1)}(1{:}2k)$, i.e., the first $2k$ bits of $r^{(1)}$. Thus in steps of k, the attacker finds the value of $r^{(1)}$ completely, and uses it to find the keys K_1 and K_2 and the unique identifier ID.

3.4 Simulations

There are two MATLAB code files (links can be found in appendix): ‘data.m’ and ‘attack.m’.

The code ‘data.m’ generates the m eavesdropped values of A, B and C

based upon randomly generated values of ID, $IDS^{(1)}$, $K_1^{(1)}$, $K_2^{(1)}$ and $r^{(1)}$ $tor^{(m)}$. Bit size is kept at 48 bits. These random values can be changed to any 48 bit number before being used to calculate *A*, *B* and *C*. Extension to 96 bit size can be obtained by running the algorithm twice. Once for bits 1 to 48 and the second time for bits numbered 49-96.

The code 'attack.m' carries out the passive attack on the eavesdropped data. The targeted variable is $r^{(1)}$, because once this is known, equations (3.1), (3.2) and (3.3) can be used to find K_1, K_2 and ID. Following the theory outlined in the paper [3], the first *k* bits of $r^{(1)}$ are guessed. Since this guessing is based upon exhaustive search on 2^k data values for finding the candidates meeting the required criteria, the choice of *k* may be limited by computational power available. Hence, in this work, *k* has been chosen as <number of bits/4>, i.e., *k*= 48/4 = 16.

The algorithm proceeds as given in Section 3.3and shown in Fig. 3.2. A small number of guesses of $r^{(1)}(1{:}k)$, is selected, which meet the required criteria in full. Second exhaustive search is run for $r^{(1)}(k{+}1{:}2k)$, using the concatenated term $r^{(1)}(1{:}2k)$,as given in equation (3.7), for each of the guesses short-listed for $r^{(1)}(1{:}k)$, in the previous step.

$$r^{(1)}(1{:}2k)=r^{(1)}(1{:}k)||r^{(1)}(k{+}1{:}2k) \qquad (3.7)$$

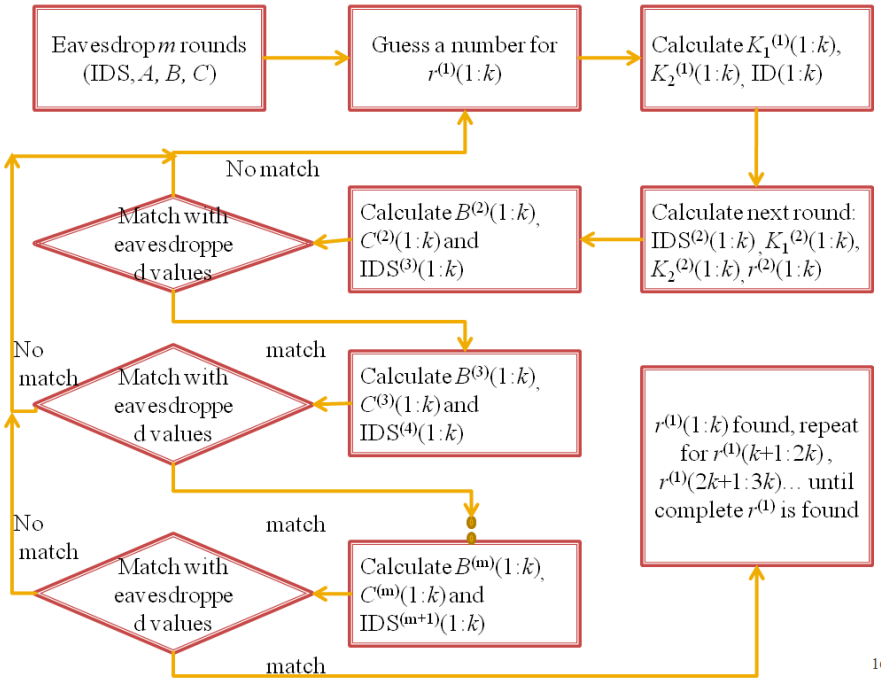


Figure 3.2: Simulation Steps for the Attack on LMAP^{++} Protocol

Thus in total 4 exhaustive searches are run on 2^k data values rather than full brute exhaustive search of 2^{4k}. This is compared through equations (3.8) and (3.9)

$$4 \times 2^k = O(2^k) \quad (3.8)$$

$$1 \times 2^{4k} = O(2^{4k}) \quad (3.9)$$

3.5 Experimental Results

The algorithm was run 100 times but the data generation was done randomly. So even though it was run 100 times, there was a chance of repetition of same set of eavesdropped parameters. This probability is very slim since there are a total of (m+4) 48-bit variables, set randomly in data generation. These are:

- r:m values $r^{(1)}, \ldots, r^{(m)}$
- ID
- IDS$^{(1)}$
- $K_1^{(1)}$
- $K_2^{(1)}$

The number of rounds m has been varied from 5 to 30, to see if it results in any change in the number of guesses. However, irrespective of m from 5 to 30, the algorithm gives two results of secret number (unless both guesses are the same, in which case one is omitted), both satisfying the eavesdropped criteria but one of these is true and the other is false.

The number of bits n was varied from 12 to 48. It provided correct guess of $r^{(1)}$ all the times. On almost all occasions, it also resulted in an additional incorrect value of $r^{(1)}$, which also met the selection criteria fully. Hence, the algorithm reduced the number of guesses from 2^n to 2, of which one is correct. These guessed values are stored in variable 'record4' in the code.

3.6 Conclusion

In this way, the work presented in [3] on LMAP^{++} disclosure attack has been replicated and verified by our simulations on 48 bit data. The result gives two values of $r^{(1)}$, out of which one is the correct guess, irrespective of the number of rounds between 5-30,. Hence the number of guesses is reduced from 2^n for n bit data to just 2 by using this attack algorithm. Different approaches may also be tried out by using values of bit search size k, other than $n/4$ used in this work.

As highlighted in earlier work [3], the attack was successful because the operators used in the protocol (XOR and modular addition), do not provide diffusion of the data bits. As a result, any data bit of the input variables effects only the bits which are of same indices or to the left of that bit with higher indices.

As we will see in Chapter 4, protocols have been developed to provide diffusion effect through rotation operator, in order to overcome this weakness.

CHAPTER 4

ATTACKING GOSSAMER RFID AUTHENTICATION PROTOCOL

4.1 Introduction

In Chapter 3, an overview of LMAP^{++} RFID authentication protocol proposed in 2008 [2] and a related passive attack in 2012 [3] was given. The attack was successful in extracting the secret value, based upon eavesdropped messages between the tag and the reader.

SASI [45] and Gossamer [4] protocols have been proposed respectively in 2007 and 2009 to enhance the security of LMAP^{++} through rotation and MIXBITS functions. However, attacks against these two protocols have also been attempted, in 2008 [46] against SASI and in 2009 [47] and in 2010 [5] against Gossamer. In this chapter, focus has been given to highlight weaknesses in a passive probabilistic attack, described by E. G. Ahmed in [5]. Also, a type of continuous Denial of Service (DoS) attack is proposed against the tag operating on SASI and Gossamer protocol.

The chapter is organized as follows. In Section 4.2, an overview of SASI and Gossamer protocols is provided. In Section 4.3, the probabilistic passive attack is discussed, as described in [47] and [5]. Section 4.4 highlights the weaknesses discovered in this attack. In Section 4.5, the proposed DoS attack is explained. The chapter is concluded in Section 4.6.

4.2 SASI and Gossamer Protocol Overview

4.2.1 SASI Protocol

In 2007, Chien proposed a very interesting lightweight authentication protocol providing Strong Authentication and Strong Integrity (SASI) for low-cost RFID tags [45]. The main difference between LMAP and SASI is the inclusion of a non-triangular function, in the form of rotation. ROT(x,y) is a circular shift of x, wt(y) positions to the left where wt(y) denotes the Hamming weight of y.

An index-pseudonym (IDS), the tag's private identification (ID), and two keys (k_1,k_2) are stored both on the tag and in the back-end database. Simple bitwise XOR ($\oplus$), bitwise AND ($\wedge$), bitwise OR ($\vee$), addition modulo 2^N (for N number of bits) and left rotation (ROT(x,y)) are required on the tag. Additionally, random number generation (i.e., n_1 and n_2) is required on the reader. The protocol is divided into three states: tag identification, mutual authentication and updating phase.

In the identification phase, the reader sends a "hello" message to the tag, and the tag answers with its IDS. The reader then finds the ID and keys, k_1 and k_2, from its database corresponding to that particular IDS, and then initiates the mutual authentication. In this, the reader and the tag authenticate each other, and the index-pseudonym and the keys are subsequently updated. Details are shown in Fig. 4.1.

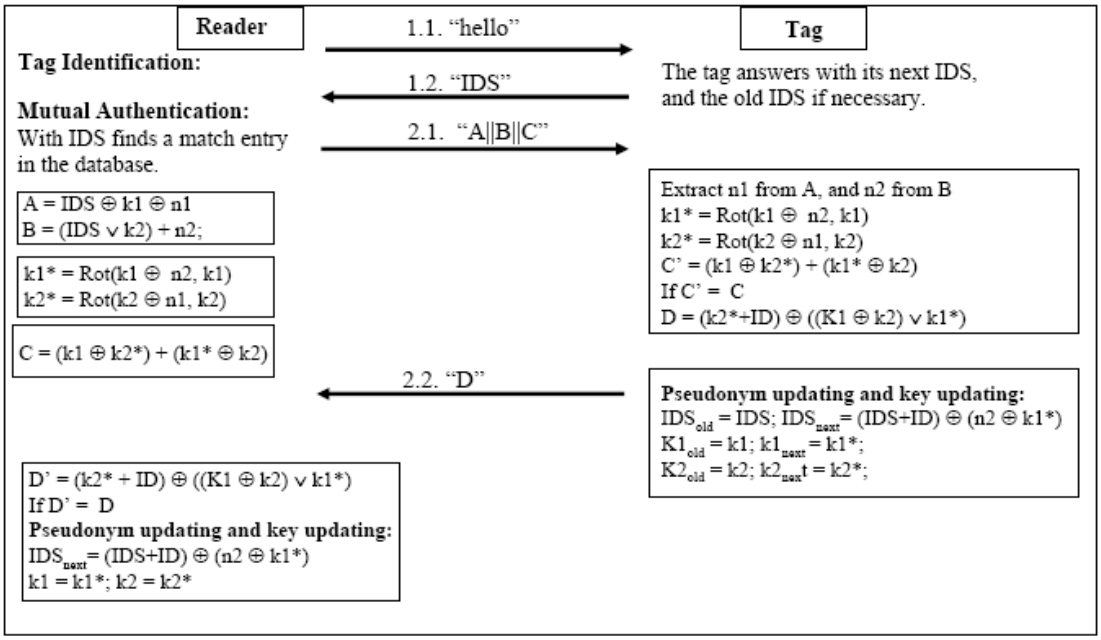


Figure 4.1: SASI (Strong Authentication and Strong Integrity) Protocol

4.2.2 Gossamer Protocol

The Gossamer scheme is similar to SASI with two differences in the calculation of the parameters. ROT(x,y) performs a left circular shift on the value of x, (y mod N) positions for a given value of number of bits (N). A lightweight function called MIXBITS is included, containing bitwise right shift (>>) and additions mod 2^n. MIXBITS operation is defined as follows.

Z = MIXBITS(X,Y) (4.1)

$Z = X$;

for (i=0; i<32; i=i+1)

{$Z = (Z>>1) + Z + Z + Y$;} ,

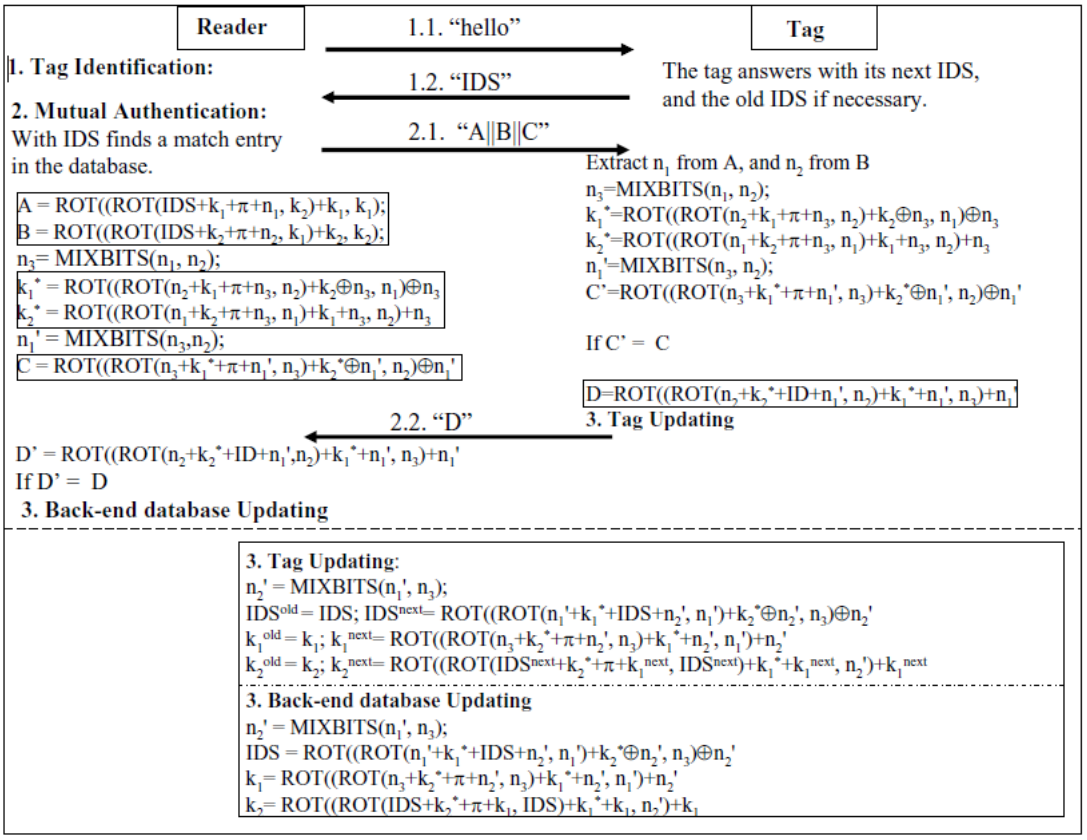
Where Z>>1 implies a right circular shift by 1. This means that we are carrying out the addition operation 32 times, and at any ith state,

$Z(i)= (Z(i\text{-}1) >> 1) + Z(i\text{-}1) + Z(i\text{-}1) + Y$ (for i = 1:32)

And $Z(0) = X$.

The protocol proceeds in three stages as follows:

The reader first sends a “hello” message to the tag, which answers with its potential next IDS. With it, the reader tries to find an identical entry in the database. If this search succeeds, the mutual authentication phase starts. Otherwise the reader requests the old IDS from the tag and its match is searched in the reader database. The mutual authentication and updating phases are carried out as shown in Fig. 4.2, where $k_1$andk_2 are the mutually known secret keys and n_1 and $n_2$are nonces, randomly selected by the reader. Variables *A*, *B*, *C*, *D*, k_1*, k_2*, n_3 and n_1'are calculated and updated as shown in Fig. 4.2.



† π = 0x3243F6A8885A308D313198A2 (*L*= 96 bits).

Figure 4.2: Gossamer Protocol.

4.3 Probabilistic Attack

Based upon the definition of ROT(x,y) operator stated in Section 4.2.1, it is reasoned in [47] that ROT(x,y) can perform 0 to 95 rotations for a 96-bit number. So if all rotations are equally probable, the probability of zero rotation is 1/96. Using this theory that one out of every 96 number gives no rotation, the probabilistic attack described in [5] is as follows:

Looking at the equation for C in Gossamer protocol given by Fig. 4.2,

$$C = \text{ROT}((\text{ROT}(n_3 + k_1^* + \pi + n_1', n_3) + k_2^* \oplus n_1', n_2) \oplus n_1' \quad (4.2)$$

where $n_3 = \text{MIXBITS}(n_1, n_2)$ and $n_1' = \text{MIXBITS}(n_3, n_2)$.
If $n_1 = 0 \bmod 96$ (i.e, it gives zero rotation) and also $n_2 = 0 \bmod 96$ (the probability of this happening is $1/96 \times 1/96 = 1/9216$), then [5] states that n_3 and n_1' are also 0 mod 96 since MIXBITS(0mod96, 0mod96) = 0mod96.

Based upon this reasoning, the authors simplify equation (4.2) to

$$C = k_1^* + \pi + k_2^*. \quad (4.3)$$

Similarly the equation for D in the protocol is

$$D = \text{ROT}((\text{ROT}(n_2 + k_2^* + \text{ID} + n_1', n_2) + k_1^* + n_1', n_3) + n_1' \quad (4.4)$$

and it is simplified to

$$D = k_1^* + \text{ID} + k_2^*. \quad (4.5)$$

The original equation for IDS^{next} is

$$\text{IDS}^{\text{next}} = \text{ROT}((\text{ROT}(n_1' + k_1^* + \text{IDS} + n_2', n_1') + k_2^* \oplus n_2', n_3) \oplus n_2' \quad (4.6)$$

This is simplified to

$$\text{IDS}^{\text{next}} = k_1^* + \text{IDS} + k_2^*. \quad (4.7)$$

Equations (4.3), (4.5) and (4.7) are solved simultaneously to get

$$C - \pi = \text{IDS}^{\text{next}} - \text{IDS}. \quad (4.8)$$

Thus if two successive exchanges satisfy (4.8), then ID is determined.

A second similar attack is also given in [5], based upon k_1 and k_2 being 0 mod 96.

4.4 Weaknesses

It has been found in this thesis that the above mentioned attack in fact does not work for all values of n_1 and n_2 due to the reasons explained below.

4.4.1 MIXBITS(0mod96, 0mod96) is NOT 0mod96

We notice that in general, MIXBITS (0mod96, 0mod96) is not equal to 0mod96, on the contrary to the claim made in [5]. Looking at the structure of MIXBITS operator in equation (4.1), the result of MIXBITS(X,Y) would be divisible by 96 if and only if (Z>>1) remains divisible by 96 after 32 cycles.

This fact is illustrated in the following simplest example, where Z=MIXBITS(96,96) as defined by (4.1). The MATLAB code and functions developed to carry out this simulation are attached in the Appendix. The calculations have been assisted in part by John D'Errico's VPI functions for handling very large integers in MATLAB [48]. We have investigated this attack under two definitions of the shift operator (>>):

Definition 1: Z>>1 implies simple right shift of Z by 1 bit with just discarding LSB and no carry-over of LSB to MSB.

Definition 2: Z>>1 implies right circular shift of Z by 1 bit with LSB shifted to MSB.

4.4.1.1 Experimental Results with Definition 1

Table 4.1 summarizes the results of Z = MIXBITS(96,96) based upon simple right shift of Z, i.e.,

Z>>1 = Z/2 (if Z is even)

Z>>1 = (Z-1)/2 (if Z is odd).

As shown in the last column, *Z*mod96 is not equal to zero.

Table 4.1: Iterations of the Operation *Z*=MIXBITS(96,96) Defined by (4.1) and Definition 1

Cycle No *i*	***Z*(*i*)=(*Z*(*i*-1)>>1)+*Z*(*i*-1)+*Z*(*i*-1)+Y (*i* = 1:32) And *Z*(0) = *X* (*X*=*Y* =*96*)**	**(Z(*i*)>>1)**	***Z* mod 96**
0	96	48 =Z/2 (Z is even)	0
1	336	168=Z/2 (Z is even)	48
2	936	468=Z/2 (Z is even)	72
3	2436	1218=Z/2 (Z is even)	36
4	6186	3093=Z/2 (Z is even)	42
5	15561	7780=(*Z*-1)/2 (*Z* is odd)	9
6	38998	19499	22
7	97591	48795	55
8	244073	122036	41
9	610278	305139	6
10	1525791	762895	63
11	3814573	1907286	13
12	9536528	4768264	80
13	23841416	11920708	8
14	59603636	29801818	20
15	149009186	74504593	2
16	372523061	186261530	53
17	931307748	465653874	36
18	2328269466	1164134733	90
19	5820673761	2910336880	33
20	14551684498	7275842249	82
21	36379211341	18189605670	13
22	90948028448	45474014224	32
23	227370071216	113685035608	80
24	568425178136	284212589068	56
25	1421062945436	710531472718	92
26	3552657363686	1776328681843	38
27	8881643409311	4440821704655	95
28	22204108523373	11102054261686	45
29	55510271308528	27755135654264	16
30	138775678271416	69387839135708	88
31	346939195678636	173469597839318	76
32	867347989196686		46

4.4.1.2 Experimental Results with Definition 2

It is seen in the first five cycles, that Z is even and small-valued, hence Z>>1 corresponds to Z/2 and there is no carry-over of LSB to MSB. However, after these five steps, the number is odd, so Z>>1 results in circular shift and it no longer remains equivalent to Z/2. Table 4.2 shows the value of Z after each of the 32 runs of the cycle. It is clear that Zmod96 is not zero.

Table 4.2: Iterations of the Operation Z=MIXBITS(96,96) Defined by (4.1) and Definition 2

Cycle No i	$Z(i)= Z(i\text{-}1)>>1 + Z(i\text{-}1) + Z(i\text{-}1) + Y$ **(for** i **= 1:32) And** $Z(0) = X$ **(**$X=Y=96$**)**	$(Z(i)>>1)$	**Z mod 96**
0	96	48 =Z/2 (Z is even)	0
1	336	168=Z/2 (Z is even)	48
2	936	468=Z/2 (Z is even)	72
3	2436	1218=Z/2 (Z is even)	36
4	6186	3093=Z/2 (Z is even)	42
5	15561	396140812571321687967719829 48	9
6	396140812571321687967720141 66	198070406285660843983860070 83	54
7	198070406285660843983860851 75	495176015714152109959650177 55	23
8	990352031428304219919323786 5	445658414142736898963685941 00	73
9	643728820428397742947550699 26	321864410214198871473775349 63	70
10	247588007857076054979977423 9	408520212964175490716718622 87	95
11	458037814535590701712714108 61	625159719839117038824076805 98	77
12	748953723767655066314065520 80	374476861883827533157032760 40	16
13	287821059133850913914284796 24	143910529566925456957142398 12	8
14	719552647834627284785711991 56	359776323917313642392855995 78	20
15	214318369301281460093400973 14	107159184650640730046700486 57	66
16	535795923253203650233502433 81	664038774197923513084470968 58	21
17	151067370419044061680596830 44	755336852095220308402984152 2	4
18	377668426047610154201492077 06	188834213023805077100746038 53	58
19	151889439976382009568290690 25	472085532559512692751865096 80	33
20	775864412512276711888446478 26	387932206256138355944223239 13	18
21	355097780995405027850237189 89	573689703069024201892838346 62	13
22	491603639917190881657873224 00	245801819958595440828936612 00	0
23	436727474650333828209243557 60	218363737325166914104621778 80	80
24	299537061483191194587669391 60	149768530741595597293834695 80	88
25	748842653707977986469173479 96	374421326853988993234586739 98	28

26	28754338398465821430205469414	143771691992329107151027347 07	38
27	71885845996164553575513673631	75557004255214445584528811983	95
28	60872371219014877548468258669	70050266866639607571006104502	45
29	33338684276140687480854721264	16669342138070343740427360632	16
30	41185481760873811085928529 20	20592740880436905542964264 60	24
31	10296370440218452771482132396	51481852201092263857410661 98	12
32	25740926100546131928705331086		78

4.4.1.3 Verification of Our Implementation of MIXBITS function

The correctness of our MIXBITS function, as per the original definition by Peris-Lopez et al. in 2008 [4] has been verified through repeated calculations with smaller numbers to facilitate understanding. The following Table 4.3 and Table 4.4 show the results of our function for X = 36 and Y = 48 and by reducing the number of bits from 96 to 6 and finding Z= MIXBITS (36,48); where 36mod6=48mod6=0.

Table 4.3: Iterations of the operation Z=MIXBITS(36,48) for 6 bit numbers using definition 1.

i	***Z*(*i*)= *Z*(*i*-1)>>1 + *Z*(*i*-1) + *Z*(*i*-1) + Y (for *i* = 1:32) And *Z*(0) = *X* (*X=36, Y =48*)**	**(Z(*i*)>>1) Definition 1:Simple right shift Z>>1= Z/2 (for even) Z>>1= (Z-1)/2 (for odd)**	**Z mod 6**
0	36	36/2=18	0
1	18+36+36+48=138=10 mod2^6	10/2=5	4
2	5+10+10+48=73=9 mod2^6	(9-1)/2=4	3
3	6	3	0
4	63	31	3
5	13	6	1
6	16	8	4
7	24	12	0
8	44	22	2
9	30	15	0
10	59	29	5
11	3	1	3
12	55	27	1
13	57	28	3
14	62	31	2

15	11	5	5
16	11	5	5
17	11	5	5
18	11	5	5
19	11	5	5
20	11	5	5
21	11	5	5
22	11	5	5
23	11	5	5
24	11	5	5
25	11	5	5
26	11	5	5
27	11	5	5
28	11	5	5
29	11	5	5
30	11	5	5
31	11	5	5
32	11		**5**

Table 4.4: Iterations of the operation *Z*=MIXBITS(36,48) for 6 bit numbers using definition 2.

i	***Z*(*i*)= *Z*(*i*-1)>>1 + *Z*(*i*-1) + *Z*(*i*-1) + Y (for *i* = 1:32) And *Z*(0) = *X* (*X*=36, *Y* =48)**	**(Z(*i*)>>1) Definition 2:Circular right shift Z>>1= Z/2 (for even) Z>>1= [1|| (Z-1)/2] (for odd)**	**Z mod 6**
0	36	36/2=18	0
1	$18+36+36+48=138=10 \bmod 2^6$	10/2=5	4
2	$5+10+10+48=73=9 \bmod 2^6$	(9-1)/2=4, Z>>1=[1 || 0 0 1 0 0]=36	3
3	38	19	2
4	15	39	3
5	53	58	5
6	20	10	2
7	34	17	4
8	5	34	5
9	28	14	4
10	54	27	0
11	55	59	1
12	25	44	1
13	14	7	2

14	19	41	1
15	63	63	3
16	45	54	3
17	0	0	0
18	48	24	0
19	40	20	4
20	20	10	2
21	34	17	4
22	5	34	5
23	28	14	4
24	54	27	0
25	55	59	1
26	25	44	1
27	14	7	2
28	19	41	1
29	63	63	3
30	45	54	3
31	0	0	0
32	48		**0**

4.4.2 X + 0 mod96 ≠ X mod2^{96}.

Apart from this, another major assumption in deriving equations (4.3), (4.5) and (4.7) is that X + 0 mod96 = X mod$2^{96.}$ This is not always true. For example, in equation (4.2), if we assume that MIXBITS(0 mod96, 0 mod96) is 0 mod96, then n_1, n_2, n_3 and n_1' are all 0 mod96, then from equation (4.2), one obtains

C = ROT ((ROT (n_3+ k_1*+π +n_1', n_3) +k_2*⊕n_1'),n_2)⊕n_1'

⇨ C = ROT ((n_3+ k_1*+π +n_1' + k_2* ⊕n_1'), n_2) ⊕n_1'

⇨ C = (n_3+ k_1*+π +n_1' + k_2* ⊕n_1') ⊕n_1'

⇨ C = n_3+ k_1*+π +n_1' + k_2*. (4.9)

Taking n_1=33^{15}×96, n_2=7^{31}×96, n_3=9^{27}×96 and n_1'=3^{45}×96, (all 0 mod 96), and random 96 bit integers k_1*andk_2*as:

k_1* = 909494701772969825406567391O,

k_2* = 394715841206963607725571488l4

Actually, n_3=MIXBITS(n_1,n_2) , which gives:

n_3 = 114245537118122223734164119 20 = 80mod96.
But equation (4.3) is obtained from (4.2) by assuming that n_3 is 0mod96, hence, with this value of n_3, we cannot move beyond equation (4.2).

Weakness 1 explained in Section 4.4.1, says that n_3 is not 0mod96. However, in order to show that there is another weakness present in the attack, which is independent from weakness 1, we assume that weakness 1 does not exist. Hence we assume that the values of n_3 and n_1' are 0mod96 in our further work to show that even if weakness 1 were not present, there is another weakness existing in the original work which would still render it ineffective.

From equation (4.9), we get C= 697044665173251532053364779 36.
Whereas, from equation (4.3), $C= k_1* + \pi + k_2*$, we get
C = 641218081510134785950239519 84.

The most we can say is that C is congruent to $k_1* + \pi + k_2*$ mod96 or that $C - (k_1* + \pi + k_2*) = 0$ mod96. (The code for carrying out the large numbers addition is attached in the Appendix.)

4.5 Proposed Denial of Service (DoS) Attack

Various active attack scenarios against Gossamer protocol are presented in [47]. A DoS attack is an attempt to make a machine or network resource unavailable to its intended users. It is an active attack since the attacker transmits fake messages and blocks legitimate ones rather than just passively eavesdropping. Looking at the similarities of SASI and Gossamer input/outputs, this attack can be implemented on both protocols. The attack that we propose against Gossamer and SASI can be described as follows.

We assume that the attacker can form a relay channel between the tag and the reader, as shown in Fig. 4.3. It can listen to messages

exchange, generate its own messages and also block the communication channel between the two when it pleases.

This attack makes use of the fact that the tag has no way of establishing that the data (*A*, *B*, *C* values, as defined in Fig. 4.2) being sent to it, is a repetitive set for a particular value of IDS.

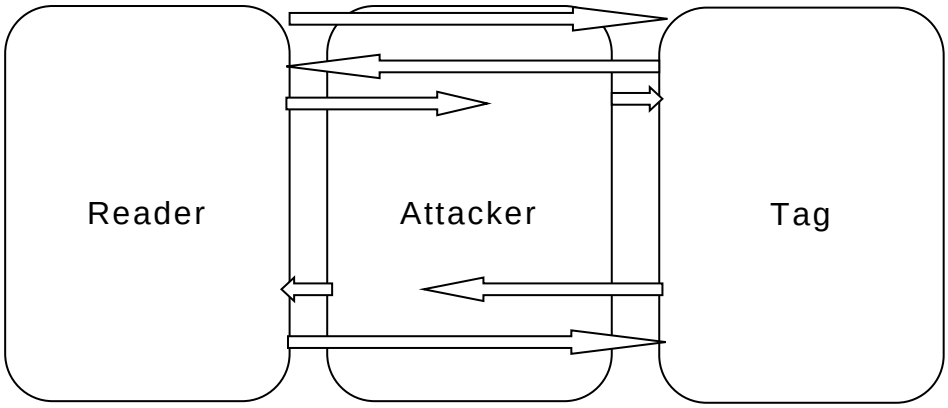


Figure 4.3: The Attacker Relay.

(a) The attacker eavesdrops on a legitimate session 1, capturing the values IDS_1, A_1, B_1, C_1, D_1, as defined in Fig. 4.2. Both the reader and the tag move to the potential new value IDS_2.

(b) The attacker now intervenes and sends hello message to the tag. The tag responds with IDS_2.

(c) The attacker claims this is not recognized and asks for the old IDS. The tag responds with IDS_1.

(d) The attacker sends the recorded data A_1, B_1, C_1 to the tag. The tag calculates C_1 and authenticates the attacker. It updates IDS_1 to IDS_2 and sends D_1 to the attacker. The attacker may choose to block D_1 from being received at the reader to prevent

giving clue about the attack. However, it does not matter to the tag under attack at present whether D_1 is received or not.

(e) The attacker repeats steps (b) to (d) continuously, as shown in Fig. 4.4, till the required time to prevent the tag from listening and replying to any legitimate query and wasting its limited computational resources on repetitive calculations.

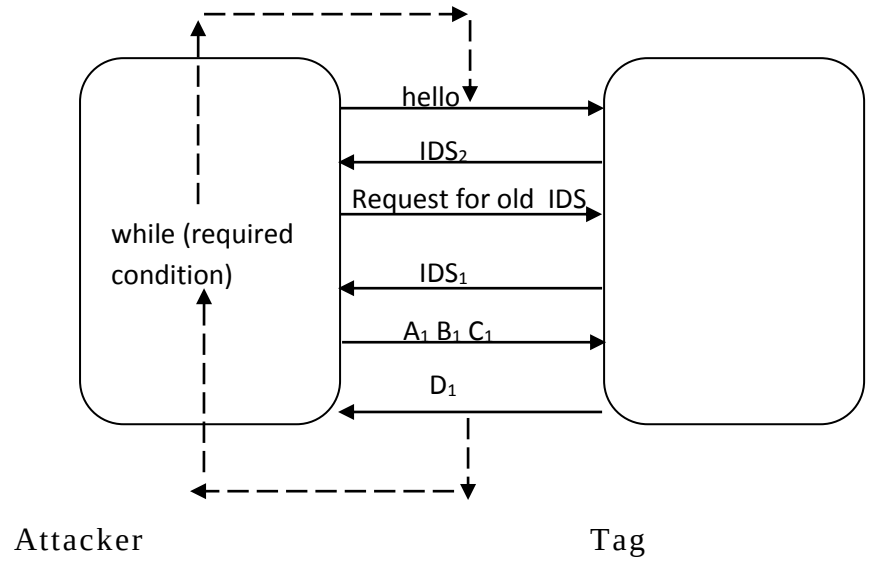


Figure 4.4: The Attack Loop.

4.6 Conclusion

The probabilistic attack presented in Section 4.4 has been investigated with two possible definitions of shift operator (>>1):

a) Simple 1 bit-shift to the right that discards the LSB

b) 1 bit circular shift to the right.

The weaknesses in the attack arise from the fact that since $96=2^5\times3$, as we repeatedly divide the number by 2, we can no longer guarantee that the number remains divisible by 96. Circular shift operator further disrupts divisibility by 2,because when numbers get large and the circular shift relocates the LSB (Least significant bit) to MSB (most significant bit), the order of the number is completely changed and thus divisibility by any specific number cannot be guaranteed.

The experiment has been repeated for smaller number of bits (10-48) to ease in understanding and same weakness has been found. The attack would have been successful if the protocol had defined left-shift in MIXBITS, instead of right-shift as a left-shift is equivalent to multiplication by 2. But if the value of nounces n_1 and n_2 were very large, then again this attack would fail due to the circular shift of MSB to LSB. These experiments have been carried out using the same code with minor changes.

The DoS attack presented in Section 4.5 makes use of the fact that since the tag is computationally small-scaled, it has no random number generation. This implies that for a particular set of {IDS, *A*, *B*, *C*}, it will always respond in a similar manner. Also, it cannot recognize that it is receiving the same parameters repeatedly. Hence the attacker forces it to carry out the same calculations again and again without recognizing the repetitions.

Unlike the DoS attack presented in [47], this attacker transmits actual eavesdropped values of *A*, *B* and *C* and not just random data. Hence it additionally forces the tag to authenticate the attacker, update its secret values and transmit *D*, substantially increasing its operation time. Thus the limited resources present on the tag are fully engaged and the tag is unable to respond to a legitimate query, till the time the attacker relay is removed from between.

CHAPTER 5

CONCLUSION

RFID is a powerful technology, which is innovating the way we live our lives. It is especially useful in managing large number of items in places where fast, automated responses are valuable such as those in retail and warehousing, access control for secure locations, baggage management at airports and livestock tracking, to name a few. Apart from its numerous benefits, RFID poses a big threat if used for malicious activities. It can help track the buying patterns and movements of persons, their social interactions and their likes and dislikes. In wrong hands, RFID communication information can be disastrous as it can cause security breaches, thefts and violation of personal privacy.

Hence, a number of light-weight proposals have been presented over time and summarized in Chapter 2 to guard the tag/reader channel from illegitimate users.

Since the light-weight protocols are restricted to the use of only basic and simple arithmetic operators (AND, OR, XOR etc), it is easier to conceive their weaknesses and counter-attacks, as was obtained for $LMAP^{++}$ protocol, discussed in Chapter 3[2,3]. In our work, the attack on $LMAP^{++}$ has been simulated and experimentally verified to be successful, along with certain newer observations.

Nevertheless, the aim of counter-attacks is only to improve the security feature, keeping in line with the principles proposed by Garfinkel [1]. The improvements on $LMAP^{++}$, in the form of SASI and Gossamer protocol are discussed in Chapter 4 along with some attacks proposed on them [4,5,45,46,47].The work conducted in this regard

finds the attack presented in [5] not to be as successful and hence the security of Gossamer is still an open question. In our work, two weaknesses have been found in the attack and experimentally verified, using two different definitions of operators implied by the Gossamer protocol architecture. These weaknesses render the attack ineffective, and the structure of the Gossamer protocol leaves little room for mounting a similar attack. However, we manage to define a new active continuous DoS attack on Gossamer in Section 4.5.

Although Gossamer provides more security than its predecessors, the inclusion of MIXBITS and ROT operators at 96 bits of data increases the tag's workload greatly and it cannot be classified under 'ultra-lightweight' protocols. For future work, a middle-man approach may be adopted to the protocol to reduce the workload of MIXBITS, while providing its highly desired diffusion property. This may be based upon a reduction of number of rounds in MIXBITS from 32 to some lower number where the required diffusion is satisfied, tested through traceability and disclosure attacks. Also, it may be looked into introducing random number generation in tag or possible memory addition to guard against our proposed attack, while keeping the cost in check.

REFERENCES

[1] S. L. Garfinkel. “Adopting Fair Information Practices to Low Cost RFID Systems”. Ubicomp 2002, September 2002.

[2] T. Li. “Employing lightweight primitives on low-cost RFID tags for authentication”. Vehicular Technology Conference, 2008.

[3] W. Shao-hui, L. Sujuan, and C. Danwei. “Efficient Passive Full-disclosure Attack on RFID Lightweight protocols”. TELKOMNIKA Indonesian J. of Elec. Eng., Vol. 10, No 6, pp. 1458-1464, October 2012.

[4] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. E. Tapiador, and A. Ribagorda.“Advances in Ultralightweight Cryptography for Low-Cost RFID Tags: Gossamer Protocol”. WISA 2008, LNCS 5379, pp.56-68, 2009. © Springer-Verlag, Berlin, Heidelberg 2009.

[5] E. G. Ahmed, E. Shaaban and M. Hashem. “Lightweight Mutual Authentication Protocol for Low Cost RFID Tags”. Int. J. of Network Security & Its Applications (IJNSA), Vol. 2, No. 2, pp.27-37, April 2010.

[6] S. E. Sarma, S. A. Weis, and D.W. Engels. “Radio-frequency identification systems”. In B. S. Kaliski Jr., Ç. K. Koç, and C. Paar, editors, CHES’02, pp.454-469. Springer-Verlag, LNCS no. 2523, 2002.

[7] S. E. Sarma, S. A. Weis, and D.W. Engels. “Radio-Frequency-Identification Security Risks and Challenges”. CryptoBytes, 6(1), 2003.

[8] S. Weis. “Security and Privacy in Radio-Frequency Identification Devices”. Master Thesis, MIT (2003).

[9] GS1 EPCGlobal Tag Data Standard. http://www.gs1.org/gsmp/kc/epcglobal, 2005.

[10] International Standards Organization. “ISO/IEC 15693: Identification cards – Contactless integrated circuit(s) cards - Vicinity cards”. http://www.iso.org, 2000.

[11] International Standards Organization. “ISO/IEC 14443: Identification cards – Contactless integrated circuit(s) cards – Proximity cards”. http://www.iso.org, 2008.

[12] A. Mitrokotsa, M. R. Rieback, A. S. Tanenbaum. “Classification of RFID Attacks”. In Proc. of 2nd IWRT’08, 10th Int. Conf. on Enterprise Information Systems, 2008.

[13] J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway. “UMAC: Fast and secure message authentication”. In Advances in Cryptology - CRYPTO ’99, Lecture Notes in Computer Science, vol. 1666. Springer-Verlag, pp.216-233, 1999.

[14] K. Yüksel, J.P. Kaps, and B. Sunar. “Universal hash functions for emerging ultra- low-power networks”. In Proc. of CNDS'04, 2004.

[15] S. Weis. “Security parallels between people and pervasive devices”. In Proc. of PERSEC’05, pp.105-109. IEEE Computer Society Press, 2005.

[16] A. Juels, R. Rivest, and M. Szydlo. “The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy”. In Proc. of CCS’03, pp.103-111. ACM Press, 2003.

[17] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. “EMAP: An efficient mutual authentication protocol for low-cost RFID tags”. In Proc. of IS’06, volume 4277 of LNCS, pp.352-361. Springer-Verlag, 2006.

[18] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. “LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags”. Hand. of RFIDSec’06, 2006. International Journal of Network Security & Its Applications (IJNSA), Volume 2, Number 2, April 2010.

[19] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. “M2AP: A minimalist mutual authentication protocol for low-cost RFID tags”. In Proc. of UIC’06, volume 4159 of LNCS, pp.912-923. Springer-Verlag, 2006.

[20] R. Rivest. The MD5 message-digest algorithm. “Request for Comments (RFC) 1321”. Internet Activities Board, Internet Privacy Task Force, April 1992.

[21] National Institute of Standards and Technologies. “Secure Hash Standard”. Federal Information Processing Standards Publication, FIPS-180, May 1993.

[22] National Institute of Standards and Technologies. “Secure Hash Standard”. Federal Information Processing Standards, Publication FIPS-180-1, April 1995.

[23] A. Sotirov, M. Stevens, J. Appelbaum, A. Lenstra, D. Molnar, D. A. Osvik, B. de Weger. “MD5 considered harmful today: Creating a rogue CA certificate”. December, 2008.

[24] F. Chabaud, A. Joux. “Differential Collisions in SHA-0”. CRYPTO’98, pp.56-71, 1998.

[25] E. Biham, R. Chen. “Near-Collisions of SHA-0”. Cryptology ePrint Archive, Report 2004/146, 2004 (appeared on CRYPTO 2004), IACR.org.

[26] X. Wang, H. Yu and Y. L. Yin. “Efficient Collision Search Attacks on SHA-0”. CRYPTO’05, CMU.edu, 2005.

[27] X. Wang, Y. L. Yin and H. Yu. “Finding Collisions in the Full SHA-1”. Crypto’05, MIT.edu, 2005.

[28] G. Bertoni, J. Daemen, M. Peeters and G. V. Assche. “The Keccak Reference”. http://keccak.noekeon.org, January, 2011.

[29] E.Y. Choi, S.M. Lee, and D.H. Lee. “Efficient RFID authentication protocol for ubiquitous computing environment”. In Proc. of SECUBIQ'05, 2005.

[30] T. Dimitriou. “A lightweight RFID protocol to protect against traceability and cloning attacks”. In Proc. of SECURE-COMM'05, 2005.

[31] J. Yang, J. Park, H. Lee, K. Ren, and K. Kim. “Mutual authentication protocol for low-cost RFID”. E-crypt Workshop on RFID and Lightweight Crypto, 2005.

[32] M. Ohkubo, K. Suzuki, and S. Kinoshita. “Cryptographic approach to privacy-friendly tags”. In Proc. of RFID Privacy Workshop, 2003.

[33] A. Juels. “Minimalist cryptography for RFID tags”. 2003. http://www.rsasecurity.com/rsalabs/staff/bios/ajuels/publications/minim%alist/index.html

[34] D. Molnar and D. Wagner. “Privacy and Security in Library RFID: Issues, Practices, and Architectures”. In Proc. of CCS’04, pp.210-219. ACM Press, 2004.

[35] D. Molnar, A. Soppera, and D. Wagner. “A Scalable, Delegatable Pseudonym Protocol Enabling Ownership Transfer of RFID Tags”. In Proc. of SAC’05, LNCS 3897, pp.276-290. Springer-Verlag, 2005.

[36] M. Aigner and M. Feldhofer. “Secure Symmetric Authentication for RFID Tags”. Telecommunication and Mobile Computing, March 2005.

[37] M. Feldhofer, M. Aigner, and S. Dominikus. “An Application of RFID Tags using Secure Symmetric Authentication”. In: Proc. of SecPerU’05, IEEE Computer Society Press, 2005.

[38] N. J. Hopper and M. Blum. “Secure Human Identification Protocols”. In Advances in Cryptology - ASIACRYPT, volume 2248 of Lecture Notes in Computer Science, pp.52-66, 2001.

[39] O. Goldreich. “Foundations of Cryptography”. Chapter 2.2.4.2, p.41. Cambridge University Press, 2000.

[40] A. Blum, A. Kalai, and H. Wasserman. “Noise-Tolerant Learning, the Parity Problem, and the Statistical Query Model”. Journal of the ACM, 50(4):506-519, July 2003.

[41] T. Li and R. Deng. “Vulnerability Analysis of EMAP-An Efficient RFID Mutual Authentication Protocol”. International Conference on Availability, Reliability and Security, vol. 0, pp. 238-245, 2007.

[42] M. Barasz, B. Boros, P. Ligeti, K. Loja, and D. Nagy. “Breaking LMAP”. In Conference on RFID Security, Malaga, Spain, July 2007.

[43] M. Barasz, B. Boros, P. Ligeti, K. Loja, and D. Nagy. “Passive Attack against the M2AP Mutual Authentication Protocol for RFID Tags”. In First International EURASIP Workshop on RFID Technology, Vienna, Austria, September 2007.

[44] T. Li and G. Wang. “SLMAP - A secure ultra-lightweight RFID mutual authentication protocol”. In Chinacrypt’07, pp.19–22, 2007.

[45] H. Chien. “SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity”. IEEE Trans. On Dependable and Secure Comp Vol 4, No 4, pp. 337-340, October 2007.

[46] J.C. Hernandez-Castro1, J. M. E. Tapiador, P. Peris-Lopez, T. Li, and J. J. Quisquater. “Cryptanalysis of the SASI Ultralightweight RFID Authentication Protocol with Modular Rotations”. eprint arXiv:0811.4257.

[47] Z. Bilal, A. Masood and F. Kauser. “Security Analysis of Ultra-lightweight Cryptographic Protocol for Low-cost RFID Tags: Gossamer Protocol”. Int. Conf. on Network-Based Information Systems, 2009 DOI 10.1109, 2009.

[48] J. D’Errico’s VPI arithmetic functions for MATLAB. (link:-http://www.mathworks.com/matlabcentral/fileexchange/22725-variable-precision-integer-arithmetic

APPENDIX

MATLAB CODES

In this appendix, we give the online links to the MATLAB codes written to verify the attack on LMAP^{++}, as discussed in Chapter 3 and the codes written for examining the weaknesses of Gossamer attack, in reference with Chapter 4.

data.m - Code for Data Setup in LMAP^{++} Attack
https://drive.google.com/file/d/0B1vzAyyEsfZWNE9oRTlvTVJ0cEE/edit?usp=sharing

---

attack.m – Code for Running the Attack on LMAP^{++}
https://drive.google.com/file/d/0B1vzAyyEsfZWVDNUUXc2dERLVVk/edit?usp=sharing

---

Code For Examining Weakness 1 in Gossamer Attack
https://docs.google.com/document/d/1cB7Hp7BIP1RZ7dIO-eyfSVxTac-hP0QRxy0EirpGv7I/edit?usp=sharing

---

Code For Examining Weakness 2 in Gossamer Attack
https://docs.google.com/document/d/1RfFZXFvS746WqxyDW-tQ6ecSxlzhoHJv49ujylmjzSM/edit?usp=sharing