Security and Privacy in Radio-Frequency Identification Devices

by

Stephen August Weis

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
[June 2003]
May 2003

Author .....
Department of Electrical Engineering and Computer Science
May 9, 2003

Certified by .....
Ronald L. Rivest
Viterbi Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by.....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Security and Privacy in Radio-Frequency Identification Devices

by

Stephen August Weis

Submitted to the Department of Electrical Engineering and Computer Science
on May 9, 2003, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science

Abstract

Radio Frequency Identification (RFID) systems are a common and useful tool in manufacturing, supply chain management and retail inventory control. Optical barcodes, another common automatic identification system, have been a familiar packaging feature on consumer items for over years.

Due to advances in silicon manufacturing technology, RFID costs have dropped significantly. In the near future, low-cost RFID "electronic product codes" or "smart-labels" may be a practical replacement for optical barcodes on consumer items. Unfortunately, the universal deployment of RFID devices in consumer items may expose new security and privacy risks not present in closed manufacturing environments.

This thesis presents an introduction to RFID technology, identifies several potential threats to security and privacy and offers several practical proposals for efficient security mechanisms. We offer several policy suggestions and discuss various open questions and areas of research.

Thesis Supervisor: Ronald L. Rivest
Title: Viterbi Professor of Electrical Engineering and Computer Science

Acknowledgments

- Thanks to my family, Joseph, Elaine, and Erica Weis, for all their love and support.
- Thanks to my advisor Ronald Rivest for his input and guidance.
- Thanks to Sanjay Sarma for the introduction to RFID and his enthusiastic support.
- Thanks to Daniel Engels for answering many of my questions.
- Thanks to the members of the Auto-ID center who have made this work possible, notably Peter Cole and Tom Scharfeld.
- Thanks to Ari Juels and Markus Jakobsson for their input.
- Thanks to Simson Garfinkel for his comments.
- Thanks to Susan, Chris, Hanson, Grant, Adam, Abhi, Matt, David, Claire and other members of MIT's LCS and AI groups for their input and friendship.
- Thanks to John Kubiatowicz and Satish Rao for their letters of recommendation.
- Props to the A-Team: Chris Gorog, Pawel Krewin, Noah Zaitlen and Tony Lobay.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Technology is a key component in managing the flow of goods and products, yet a gap still exists between the digital and physical worlds. While data abstractions may represent physical objects, those abstractions have no connection to the real world. An entry in a database indicating an item is stored in a particular location is nothing more than a snapshot taken at the moment of last human intervention. If an object is moved, the database is no longer accurate. Eventually someone still needs to physically verify that the object is present.

This is beginning to change. New technology is enabling the automatic identification, or *auto-ID*, of physical objects. Auto-ID is a core component of automated inventory control systems and supply chain management. Inventories once taken by hand will be conducted automatically. Continual database updates will better reflect the real world. Essentially, snapshots will be replaced by "live video".

One day it is conceivable that every man-made object will be labeled with a unique identity associated with a digital entity. An object's history, ownership, or location may all be available online. Each object will have its own "bitmass" [73] – some amount of digital data associated with it. Considering an object's bitmass may become as important as its weight or volume.

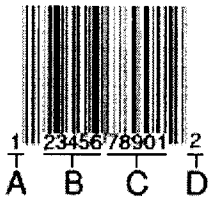

Figure 1-1: A standard UPC. Code components are labeled as follows: (A) Application Code (B) Manufacturer Code (C) Product Code (D) Checksum Digit

1.1 History of Auto-ID

Object identification became a necessity with the rise of trade and transport. A simple label allowed traders to identify packages without having to individually inspect each package. Aggregate inventory data in a cargo manifest provided an efficient means of accounting. In fact, some of the earliest forms of human writing consisted of accounting and inventory records recorded on cuneiform tablets. The ability to quickly determine the contents of a package lowered costs and transport time.

As the volume and variety of trade goods exploded in the 20th century, logistics and inventory control costs began to mount. Food chains and supermarkets were heavily affected by these costs. Supermarkets were among the first to push for development of efficient means of automatically identifying products. In 1949, a graduate student at the Drexel Institute of Technology named Norman Woodland was intrigued by the idea and tackled the problem.

Woodland related the problem to Morse code. Morse coded messages consisted of simple "dots and dashes" which could be read automatically or by humans. Legend has it that while pondering the problem at a beach, Woodland wrote a Morse code message in the sand, then extended the dots and dashes downward – producing the thick and thin lines which appear in the now familiar barcode, pictured in Figure 1-1.

Barcodes of various forms were developed over then next 20 years. In 1969, a consortium of food distribution trade associations called the Uniform Code Council (UCC) [102] began developing a standardized barcode for consumer items, dubbed the Universal Product Code (UPC) [102]. The UPC is a linear, or one-dimensional, barcode containing

manufacturer and brand information, but no unique identifying data, like the barcode in Figure 1-1. By 1974, the UCC had agreed upon a UPC standard and developed the requisite technology. On June 26, 1974, a UPC-labeled 10-pack of Wrigley's gum was scanned in an Ohio supermarket and ushered in the modern age of consumer product auto-ID.

Optical barcodes are ubiquitous in consumer retail and appear on nearly every commercial item. Two-dimensional barcodes, which can carry more data in a smaller surface area, are now frequently used by shipping and transit companies, such as UPS, Federal Express and the United States Postal service. Consumers may even print their own two-dimensional barcode postage stamps [98], possibly including cryptographic properties [101].

Optical barcodes suffer from several flaws. Objects must be physically manipulated to align barcodes with scanners. Barcodes may be smudged or obscured by shrink wrap, decreasing efficiency. Anyone who has shopped in a market has likely witnessed a checker struggling to scan an item. Retailers also often affix their own needlessly redundant barcodes on top of existing packaging. These issues limit the performance of optical barcode based auto-ID systems.

1.2 Radio Frequency Identification

One auto-ID system lacking the flaws of optical barcodes is based on radio frequency identification (RFID). The term "RFID" could be applied to systems in use for more than sixty years. Perhaps the first radio identification technology was the "Identify Friend or Foe" system used in Allied aircraft during World War II [88]. In early 1940, the British Royal Air Force outfitted airplanes with radio transponders that would respond when interrogated. This allowed pilots and ground crews to distinguish the RAF airplanes from the Luftwaffe's, which proved to be a decisive advantage in the Battle of Britain.

RFID transponders, or *tags*, carry object identifying data. This data may include the manufacturer, brand, model and a unique serial number. Collectively, this data is often referred to as the tag's identity, or ID. An ID may be of any length. In practice, a 96 bit ID would suffice for most applications.

RFID tags consist of a small microchip attached to an antennae or other coupling ele-

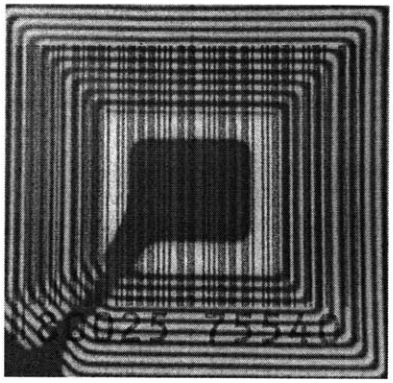

Figure 1-2: An RFID tag with a printed barcode from Checkpoint Systems.

ment. The tag communicates via radio frequencies (RF) with a transceiver, or *tag reader*. The tag ID may be read automatically: without line of sight, through non-conducting material such as cardboard or paper, at a rate of several hundred reads per second and from a distance of several meters.

Since tags typically are silicon-based microchips, functionality beyond simple identification may be incorporated into tag designs. This functionality might range from integrated sensors to read/write storage to encryption and access control support. An example tag with a printed barcode is depicted in Figure 1-2. Figure 1-3 shows a vial filled with thousands of dust-sized RFID microchips suspended in a liquid.

RFID systems have emerged as a practical auto-ID platform in industries as varied as automobile manufacturing, microchip fabrication and even cattle herding. The latter example is actually one of the first commercialized RFID systems [60]. A rugged RFID tag with a unique ID was attached to each cow's ear, allowing herders to track a particular animal as well as take temperature readings. These tags could have also contained vaccination records or any other special information (e.g. "this cow is kosher"). This offers a great advantage over traditional animal identification such as collars, tattoos or branding.

The potential benefits of a pervasive RFID system are enormous. Worldwide, over five billion barcodes are scanned daily [33]. However, barcodes are typically scanned only once during checkout. Manufacturers, transport companies and retailers may each use their own incompatible auto-ID systems. To offer insight into the potential size of the RFID market,

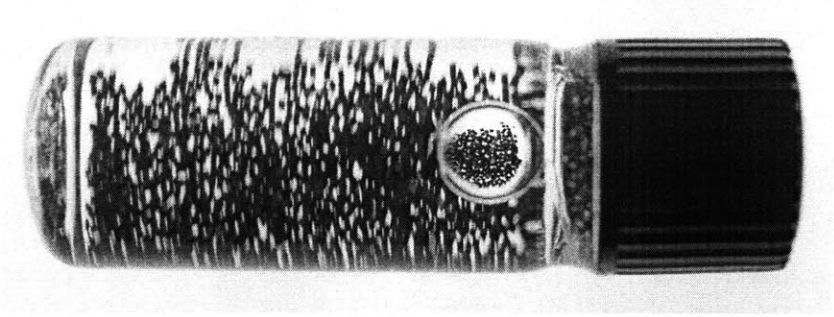

Figure 1-3: A vial filled with thousands of dust-sized RFID microchips from Alien Technologies.

Table 1.1 estimates the size of several selected companies' supply chains.

By integrating a unified identification system on all levels in the supply chain, every party involved in the lifespan of a product may reap the benefits of an RFID-based object identification system. This includes not only manufacturers and retailers, but also consumers, regulatory bodies such as the United States Food and Drug Administration, and even waste disposal or recycling firms. RFID-enabled systems may greatly lower the cost of supply chain management, inventory control and retail checkout. In fact, the aggregate savings are so great that RFID tags will likely become one of the most widely deployed microchips in history.

1.3 RFID Applications

To illustrate an RFID system, consider the application of an RFID-enhanced warehouse. Each item in the warehouse would be labeled with an RFID tag containing identifying data, such as a manufacturer code, product type and a unique serial number. In practice, the tag contents (i.e. its ID) could be represented by a 96 bit number.

Within the RFID-enhanced warehouse, shelves, forklifts and doorways would each be equipped with RFID reading devices. Shelves would "know" their contents and would recognize when items were added or removed. Similarly, forklifts would know what they were carrying and doors would know which items passed through. Each of these trans-

Company	Units (billions)
CHEP	0.2
Johnson & Johnson	3.0
Kimberly Clark	10.0
WestVaco	10.0
Gillette	11.0
YFY	15.0
Tesco	15.0
Proctor & Gamble	20.0
Unilever	25.0
Altria	30.0
Wal-Mart	53.0
International Paper	53.0
Coca-Cola	200.0
Sub-Total	412.2
(Over-counting 15%)	-61.8
US Postal Service	200.0
Total	555.4

Table 1.1: Estimated Units in Supply Chain for Selected Companies

actions could be recorded in a unified database, creating a detailed account of the entire history of any particular item. External data such as product information or sales records could also be associated with each item.

Consider the advantages offered by an RFID warehouse. Inventories, once taken periodically by hand, could be automatically updated in real time. At any moment, any item in the warehouse could be automatically located. Someone could move an item without having to record that fact - the shelves would record it for them.

Automatic analysis of movement and clustering patterns could aid in optimizing warehouse layouts. Finally, constant inventory monitoring could greatly reduce inventory "shrinkage" (an industry euphemism for theft). This represents a significant cost savings for manufacturers. Stocks of consumer items like disposable razors "shrink" by as much as 15% before even reaching a retail store. Once in the store, high-theft items like razors or cigarettes are often kept in special cases which discourage sales. The costs due to shrinkage were so great that a major razor manufacturer recently started tagging their products [78].

Many other RFID applications may emerge. Consider an airport setting. Both boarding passes and luggage labels could be tagged with RFID devices. Before take-off, an RFID-enabled airplane could verify that all boarding passes issued were on the plane and that all luggage associated with those passes was in the hold. Within an airport, tracking passengers by their boarding passes could improve both security and customer service. Of course, in other environments this would be an undesirable violation of privacy.

Self-aware storage could simplify baggage claim. A passenger could query the airport system for the whereabouts of her luggage. Extending this idea to other applications, one can envision self-aware lockers, medicine cabinets, refrigerators or laundry machines.

A variety of innovative applications may arise if consumers are allowed to build applications for their own tags. This is a major motivation for not disabling tags at checkout, designing an open platform that facilitates independent development and not prohibitively restricting tags through legislation.

1.4 Low-Cost RFID and Electronic Product Codes

To date, most RFID systems have been deployed for higher value items, such as cars or microchips. This allows tag costs to be in the range of US$0.50-US$1.00 or greater. At this price, a tag may possess a significant number of gates and could support strong cryptographic functionality. Furthermore, there may be fewer restrictions on the physical dimensions of these higher cost tags. They could be encased in rugged, tamper-resistant packaging and be firmly attached to products.

The most lucrative market for RFID tags is in consumer items, particularly as a replacement for the UPC. Manufacturers may only afford to adopt these simple electronic product codes (EPC) if they are priced in the range of US$0.05-US$0.10 (5-10¢). Additionally, EPC tags must be able to be incorporated into most packaging. Further references to "low-cost RFID tags" will imply simple "5¢" EPC-type tags.

Achieving these cost targets requires a system-wide approach encompassing every aspect of the RFID design. Integrated circuit (IC) design, RF protocols, reader design, back-end systems, IC manufacturing and antenna manufacturing must all be coordinated to keep

costs low.

Unfortunately, the universal deployment of low-cost RFID tags in consumer items may create new threats to privacy and security not present in closed manufacturing environments. Without proper security controls, tags embedded in consumer products could leak potentially embarrassing information. Even if tag contents are secured, predictable tag responses could be tracked, violating one's "location privacy". Insecurely tagged inventory of a retail store could be exploited by corporate spies deriving sales data.

The move toward RFID tagged consumer items has already begun. Clothing, shoe and accessory makers have all started embedding RFID tags in their products [24, 35, 79]. Consumer reaction has ranged from being completely oblivious to the presence of tags to outrage over privacy concerns, including a boycott against a clothing maker [22].

The difficulty in providing security in EPC systems lies in the tight cost restrictions. At the 5-10¢ range, tags simply lack the gate count to provide strong cryptographic functionality. Even if advancements in RFID manufacturing technology allow more gates at the same price, there will be continual pressure from manufacturers to lower tag costs.

This is due to the economics of the tag market. Tag purchases will likely be made by manufacturers in very high volumes. Even slight differentials in tag costs are greatly multiplied. Consider the recent purchase of 500,000,000 tags by a consumer products manufacturer [78]. A difference of US$0.01 represents US$5,000,000 in savings. Buyers must justify the costs of any additional functionality such as strong security mechanisms.

General low-cost RFID research is part of ongoing work at the MIT Auto-ID Center [67]. The author of this thesis presents an overview of RFID systems and their security implications in [91] and [92]. Several of the proposals appearing in this thesis were also presented by the author in [108]. Tom Scharfeld's thesis [93] contains a general introduction and cost analysis of low-cost RFID systems. Other RFID security proposals are presented by Juels and Pappu in [52] and by Juels, Rivest and Szydlo in [53].

Issues explored in the context of smart cards are most closely related to the resource-scarce environment of RFID devices. Relevant security issues are addressed in a broad range of smart card and tamper resistant hardware literature. Cost and security trade-offs of smart cards are analyzed by Abadi, et. al., in [1]. RFID tags may operate in insecure en-

vironments or be subject to intense physical attacks. An analysis of smart card operation in hostile environments is presented by Gobioff, et. al., in [43]. Weigart [107] presents a comprehensive overview of many physical attacks and countermeasures. Anderson and Kuhn detail specific low-cost physical attacks in [3]. The University of Cambridge's TAMPER Lab [100] conducts ongoing research into low-cost physical attacks.

Many results pertaining to implementations of cryptographic primitives are relevant to RFID devices. Cautionary information regarding the implementation of AES in smart cards is presented by Chari, et. al., in [19]. Being passively powered and relying on a wireless interface may make RFID devices especially susceptible to fault induction, timing attacks or power analysis attacks.

Boneh, DeMillo and Lipton discuss the importance of checking for protocol faults in [13]. Paul Kocher offers cryptanalysis using timing attacks in [59]. Kocher, Jaffe and Jun discuss differential power analysis in [58]. Kaliski and Robshaw discuss various attacks against cryptographic devices in [56]. Location privacy risks present in Bluetooth technology and relevant to RFID systems are addressed by Jakobsson and Wetzel in [51].

Chapter 2 contains a brief primer on RFID systems components, the interaction among those components and various other system issues. Chapter 3 describes the security and privacy issues which may arise in a universally deployed RFID system. Chapter 4 states assumptions about resources, performance requirements and the operating environment of current tag technology. Under these assumptions, several proposals are made in Chapter 5 for mechanisms to enhance security and privacy. Suggestions for policy decisions appear in Chapter 6. Finally, several open questions and future areas of research are outlined in Chapter 7. Appendix A contains a glossary of terms defined and used throughout this thesis. Appendix B contains a list of characters used to personify various attacks in the description of protocols.

Chapter 2

RFID System Primer

2.1 RFID System Components

RFID systems are composed of three key components:

- the RFID tag, or *transponder*, carries object identifying data.
- the RFID tag reader, or *transceiver*, reads and writes tag data.
- the back-end database stores records associated with tag contents.

2.1.1 Tags

Every object to be identified in an RFID system is physically labeled with a tag. Tags are typically composed of a microchip for storage and computation, and a coupling element, such as an antenna coil for communication. Tags may also contain a contact pad, as found in smart cards. Tag memory may be read-only, write-once read-many or fully rewritable.

A key classification of RFID tags is the source of power. Tags may come in three flavors: active, semi-passive and passive. Active tags contain an on-board power source, such as a battery, as well as the ability to initiate their own communications; possibly with other tags. Semi-passive tags have a battery, but may only respond to incoming transmissions. Passive tags receive all power from the reader and necessarily cannot initiate any communications.

	Passive	Semi-Passive	Active
Power Source	Passive	Battery	Battery
Transmitter	Passive	Passive	Active
Max Range	10 M	100 M	1000 M

Table 2.1: Active, Semi-Passive and Passive Tags

To offer an analogy for the passive powering process, one may think of readers as "shouting" out to passive tags, then extracting data from the resultant echoes. Passive tags are completely inactive in the absence of a reader. A more detailed discussion of the passive powering mechanism appears in Section 2.2.1.

A tag's power source determines both its range and cost. Passive tags are the cheapest to manufacture and incorporate into packaging, yet have the shortest read range. Semi-passive tags have moderate range and cost, while active tags have the greatest range and cost. Semi-passive and active tags' on-board power source may also power a clock or integrated sensors. Refer to Table 2.1 for a comparison of the various tag types.

It is also convenient to classify tags by their functionality. The MIT Auto-ID Center [67] has defined five classes based on functionality [4]. We offer similar classifications defined below and in Table 2.2:

Class 0: Class 0 tags are the most primitive tag, offering only *electronic article surveillance* (EAS) functionality. EAS tags only announce their presence and do not contain any unique identifying data. Class 0 tags may be "chipless" – containing no logic. They are frequently found in library books or compact discs.

Class 1: Class 1 tags contain unique identifying data stored in read-only or write-once read-many (WORM) memory. Class 1 tags will typically be passive, although may be semi-passive or active. Class 1 tags function as simple identifiers and are the focus of this thesis.

Class 2: Class 2 tags have read-write memory, which allows them to act as logging devices. Class 2 tags may be recycled and used to identify many different items throughout their lifetime. Although Class 2 could be passive, they are more likely to be semi-passive or active.

Class	Nickname	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

Table 2.2: Tag Functionality Classes

Class 3: Class 3 tags contain on-board environmental sensors. These may record temperature, acceleration, motion or radiation. To be more useful than a memoryless sensor, Class 2 tags require writable storage. Since sensor readings must be taken in absence of a reader, Class 3 tags are necessarily semi-passive or active.

Class 4: Class 4 tags may establish ad hoc wireless networks with other tags. Since they may initiate communication, Class 4 tags are necessarily active. Functionally, these tags lie in the realm of ubiquitous computers or "smart-dust" [54].

This thesis is primarily focused on Class 1 tags, also known as EPC tags. The costs and resources of Class 1 tags are stated in Chapter 4. Chapter 5 proposes mechanisms thought to be appropriate for EPC tags.

2.1.2 Readers

Tag readers interrogate tags for their data through an RF interface. To provide additional functionality, readers may contain internal storage, processing power or connections to back-end databases. Computations, such as cryptographic calculations, may be carried out by the reader on behalf of a tag.

The channel from reader-to-tag may be referred to as the *forward* channel. Similarly, the tag-to-reader channel may be referred to as the *backward* channel. The interaction between tags and readers is discussed more in Section 2.2.1.

In practice, readers might be handheld devices or incorporated into a fixed location. One application of a fixed reader is a "smart shelf". Smart shelves could detect when items are added or removed, and would play a key role in a real-time inventory control system.

Fundamentally, readers are quite simple devices and could be incorporated into mobile devices like cellular phones or PDAs. A basic reading device has been constructed for US$5 [40]. A stand-alone, hand-held reader with a wireless connection to a back-end database may cost around US$100-200. If RFID tags become ubiquitous in consumer items, tag reading may become a desirable feature on consumer electronics.

2.1.3 Back-End Database

Readers may use tag contents as a look-up key into a back-end database. The back-end database may associate product information, tracking logs or key management information with a particular tag. Independent databases may be built by anyone with access to tag contents. This allows unrelated users along the supply chain to build their own applications.

It is assumed that a secure connection exists between a back-end database and the tag reader. For protocol analysis, it may sometimes be useful to collapse the notion of reader and back-end database into a single entity. In other cases, the reader may be treated simply as an untrusted channel between tag and database.

In many ways, tags are only useful if corroborated with a database in some way. This is particularly true if tags do not contain explicit data, such as manufacturer and product codes. Tags could contain pointers, randomized IDs or encrypted data. While anyone could build a database from scratch using these values, it will often be more economical to subscribe to a database already containing tag associations.

2.2 RFID System Interface

2.2.1 Tag-Reader Coupling

Passive RFID tags receive power by harvesting energy from the electromagnetic field of a reader's communication signal. Tags must both receive power and communicate within a narrow band of radio frequencies specified by regulation agencies such as the Federal Communications Commission (FCC). We denote the center of this band of frequencies as f. When referring to RFID systems operating at frequency f, it implies that this is the

center of an operating band of frequencies.

Passive tags typically receive power through inductive coupling or through far-field energy harvesting. Inductive coupling uses the magnetic field generated by the reader to induce an electric current through a coupling element, usually an antenna and capacitor. The current from coupling charges a capacitor which provides voltage and power to the tag. Inductively coupled systems behave similar to loosely-coupled transformers. Consequently, inductive coupling only works in the near-field of the communication signal, which extends a $\frac{1}{2\pi}$ times a signal's wavelength from the source. For example, a 13.56 MHz signal has a wavelength of approximately 22 meters and will have a near-field of about 3.52 meters. Frequency regulations and restrictions on antennae size may further reduce the effective range of inductively coupled tags.

The operating voltage of an inductively coupled tag depends on the flux density at that range from the reader. At distance d, the magnetic field emitted by a reader has decreased to $\frac{1}{d^3}$ its original strength. For a circularly coiled reader antenna with radius R, the flux is maximized at distance d when $R \cong \sqrt{2}d$. Thus, increasing R increases the range of optimal communication.

Besides inductive coupling, tags may be powered by collecting energy from the far-field, which is the range outside $\frac{1}{2\pi}$ the wavelength of a signal. As with inductive coupling, the power available to a tag decreases proportional to the distance from the reader. In this case, at a rate of $\frac{1}{d^2}$.

Several challenging issues arise from both powering and communicating over the same signal. First, any modulation of the signal will reduce power to tags. Second, modulating information into an otherwise pure sinusoid spreads the signal in the frequency domain. This spread, referred to as "side band", and the maximum power of transmissions are usually regulated by local governments. These restrictions limit the amount of information which may be sent from reader to tag. The Industrial-Scientific-Medical (ISM) bands allocated to RFID devices are particular stringent in this respect.

2.2.2 Data Coding

Data stored on tags must be sent to readers in a reliable manner. Encoding this data and transmitting it over a modulated signal are two critical components of reliable communications. The choice of coding and modulation schemes determines the bandwidth, integrity and power consumption of tag-to-reader communications.

The power and modulation capabilities of tags restrict which coding and modulation schemes are appropriate for RFID systems. Bandwidth is another limiting factor. Although readers may transmit at high power, government restrictions typically limit the side band resulting from modulation. However, since passive tags do not actively transmit a signal, the encoding on the backward channel is not subject to these restrictions and may occupy a high bandwidth.

Two broad categories of codes used in RFID systems are *level codes* and *transition codes*. The former scheme represents binary values by a specific voltage level, while the latter represents bits by transitions between voltages. Level codes tend to be history independent, yet may lack robustness. Transition codes are more robust, although may be history dependent. Several coding schemes are depicted in Figure 2-1.

A simple code is Pulse Pause Modulation (PPM) in which the length between pulses represents bit values. PPM codes have narrow bandwidth and are easy to implement, but have a low bit rate. By comparison, Manchester codes represents 1's and 0's as negative and positive transitions, respectively. Both the bit rate and bandwidth of Manchester codes are higher than PPM.

The coding technique in RFID systems must be selected with three criteria:

1. The code should maintain power to the tag as much as possible.

2. The code should not consume too much bandwidth.

3. Collisions must be detected.

The first criteria favors PPM and PWM codes, because of their relatively stable signal. PPM and PWM codes also satisfy the second criteria. However, detecting collisions favors a Manchester code.

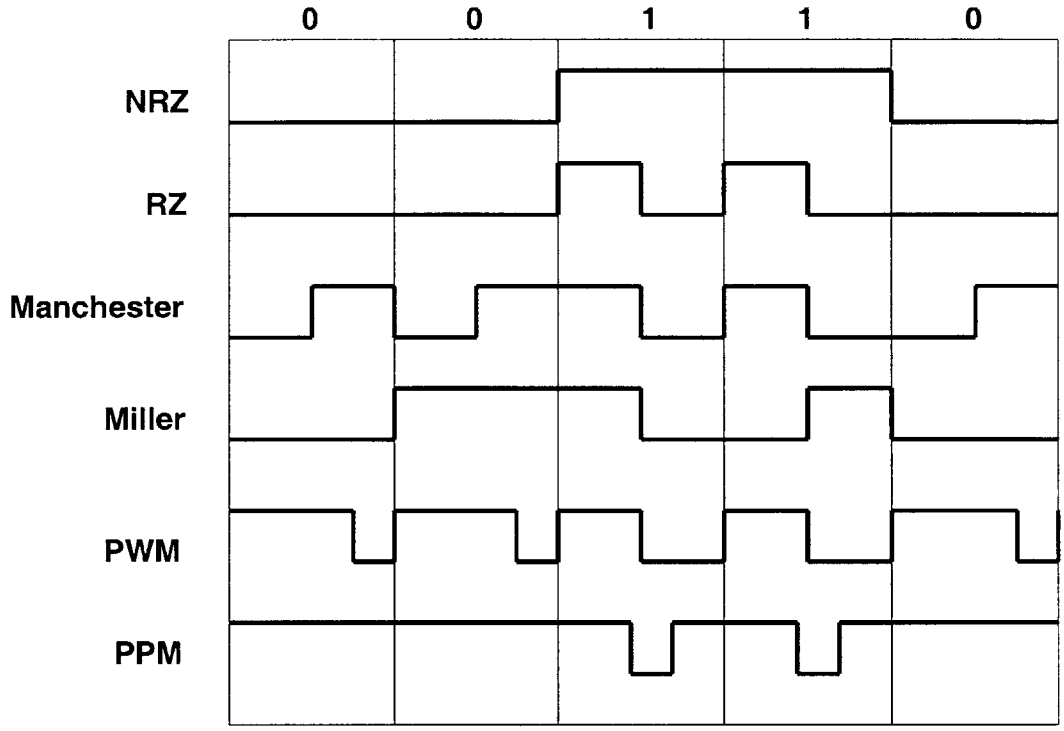

Figure 2-1: Examples of several coding schemes.

One solution is to use PPM coding on the forward channel and Manchester coding on the backward channel. This is advantageous since PPM maintains power and consumes little bandwidth, and collisions may be detected if multiple tags respond with Manchester codes. Since the backward channel's bandwidth is not subject to regulation, Manchester relatively high bandwidth is not a problem.

2.2.3 Modulation

While data coding determines the representation of data, modulation determines exactly how tags and readers communicate. RF communications typically consist of a carrier wave modulated to carry data. There are three main classes of digital modulation: Amplitude Shift Keying (ASK), Frequency Shift Keying (FSK) and Phase Shift Keying (PSK). Each class has its own power consumption, reliability and bandwidth requirements.

The vast difference in power between tags and readers creates a unique problem for RFID systems. In some differences, the return signal to the reader may be overwhelmed

by the outgoing signal, rendering tag responses impossible to detect. To prevent this from occurring, the return signal is sometimes modulated onto a different frequency, or sub-carrier. For example, in the ISO 15693 standard for 13.56 MHz RFID, a sub-carrier of 13.56 MHz/32 (= 423.75 KHz) is used [49].

2.2.4 Tag Anti-Collision

Readers may attempt to read a single tag from among a population of many. When multiple tags respond simultaneously to a reader query, conflicting communication signals may cause interference. This interference is called a *collision* and may result in a failed transmission.

Readers and tags may must employ a method to avoid collisions, referred to as an anti-collision algorithm. Similar collision problems in cellular telephone networks and Ethernet local area networks are also handled by anti-collision algorithms. RFID systems have several unique traits in regards to collisions. Computation power is restricted, tags have unreliable states or are stateless, and collisions may be difficult to avoid due to varying signal strengths. Furthermore, tags are assumed not to be able to communicate with each other. This places the full responsibility of detecting collisions on the reader.

Anti-collision algorithms may be either probabilistic or deterministic. A familiar probabilistic algorithm is the Aloha scheme [11, 66] used in Ethernet local area networks. In the tag-reader context, tags avoid collisions with other tags by randomly delaying their responses. If a collision does occur, the reader will inform all nearby tags and the culprits will wait another, usually longer, random interval before continuing. Higher densities of tags will result in a higher collision rate and degraded performance. The ISO 15693 standard for RFID supports a slotted Aloha mode of anti-collision [49].

A simple deterministic algorithm is the binary tree-walking scheme. In this scheme, a reader will query all tags in the vicinity for the next bit of their ID. If two different bit values are transmitted from among the population of tags, the reader will be able to detect the collision. The reader will then broadcast a bit indicating whether tags who broadcast a 0 or tags who broadcast a 1 should continue. Essentially, the reader chooses a "branch"

from the binary tree of ID values.

Tags which do not match the reader's choice will cease participating in the protocol. As the reader continues to move down the branches of the binary tree, fewer tags will continue operating. If all tags are unique, at the end of the protocol only a single tag will remain in operation. This process of addressing and isolating a single tag is referred to as *singulation*.

Several metrics may be used to judge the quality of anti-collision algorithms:

1. Performance.
2. Range.
3. Bandwidth requirements.
4. Implementation costs.
5. Noise and error tolerance.
6. Security.

Binary tree-walking's strengths are efficient performance and low tag implementation costs. At the end of singulation, the reader will know the entire ID of the tag it is communicating with. However, this scheme contains a threat to security. This issue is addressed further in Section 5.4.

Bandwidth regulations are a major consideration in choice of anti-collision algorithm and are discussed in the Section 2.2.5. Probabilistic algorithms tend to consume less bandwidth. As a result, tags operating in the highly-regulated 13.56 MHz band tend to use probabilistic algorithms, while tags operating in the less-regulated 915 MHz band typically use deterministic algorithms.

2.2.5 Frequencies and Regulations

Local government regulation of the electromagnetic spectrum affects the operation of RFID systems. Most RFID systems operate in the Industrial-Scientific-Medical (ISM) bands, which are freely available to low-power, short-range systems. These bands are defined by

the International Telecommunications Union (ITU) [50]. An overview of these regulations appears in [93].

In the United States, the most common ISM bands used by RFID systems are 13.56 MHz and 902-928 MHz. Low-frequency licenses are also available in the 9 kHz - 135 kHz bands. Devices operating in each band are subject to different power and bandwidth regulations, presenting different challenges to each application.

For example, systems operating in the 13.56 MHz band are limited to a bandwidth of 14 kHz in the forward channel. The backward channel may use a greater bandwidth, since it has much lower power.

In contrast, the 915 MHz ISM band is less restricted and several options are available for reader-to-tag communications. The option that provides the longest read range requires the reader to "hop" among 50 channels every 0.4 seconds, each with up to 250 kHz of bandwidth. This is a trade-off, since tags cannot be guaranteed continuous communication across a frequency hop. As a result, reader/tag communications must be limited to 0.4 seconds. Transactions must be completed within this period, otherwise they will be interrupted by a frequency hop.

Chapter 3

Security and Privacy Issues

The great efficiency gains offered by RFID systems may come at the cost of both privacy and security. Vulnerabilities to physical attacks, counterfeiting, spoofing, eavesdropping, traffic analysis or denial of service could all threaten unprotected tags.

Each of these risks may affect the privacy and security of both individuals and organizations. Following the the tradition of many cryptographic papers, a cast of human characters will represent various attacks against RFID systems. These characters are listed in Appendix B for future reference.

3.1 *Dramatis Personae*

Phyllis: Phyllis is the strongest attacker. She may conduct physical attacks against tags, such as those specified in [3] and [107]. Phyllis is assumed to be able to physically obtain tags and conduct sophisticated attacks in a laboratory setting. Her attacks may include probe attacks, material removal through shaped charges or water etching, energy attacks, radiation imprinting, circuit disruption or clock glitching.

Fortunately with the exception of TEMPEST [71] attacks, Phyllis cannot carry out her attacks in public or on a widespread scale. It is also rather moot to be concerned about privacy and security when she can surreptitiously obtain tags embedded in a product's packaging without detection.

Mallory: Mallory does not have physical tag access, but may actively participate in

protocols or construct her own counterfeit tags. Mallory may initiate queries to tags or respond to reader queries at will.

Eve: Eve plays a passive role. She cannot actively take part in protocols and is limited to eavesdropping. Eve may only listen to "logical" messages - the 1's and 0's transmitted in protocols, as opposed to the electromagnetic emissions monitored by Phyllis in TEMPEST attacks.

Tracy: Tracy is weaker than Eve. Tracy cannot read the contents of messages, but still may detect their presence. In other words, Tracy is limited to traffic analysis and may detect how many and when messages are sent. Tracy may conduct attacks against "location privacy"[9]. In some situations, Tracy may be as threatening as Eve.

Denise: Denise is the weakest of all the characters. She can neither read nor even detect the presence of messages. Denise is limited to disrupting broadcasts, blocking messages or any other denial of service attacks. As RFID becomes more mainstream, Denise's attacks may become ever more damaging.

3.2 Threats and Attacks

Any of the attacks personified by these characters may threaten security or privacy. For instance, consider retail items tagged with insecure RFID labels that are carried by consumers. If these tags lack access control, Mallory may arbitrarily query them for their contents. This might seem innocuous upon first inspection. Anyone might occasionally glance into your shopping cart or peek into your bag.

However, casual snooping cannot be done automatically at will or on a widespread scale as Mallory could. Books, magazines, medicines, underwear, birth control - these are all things we might not want arbitrary strangers to know about. Besides just nosy neighbors, scrupulous marketers or thieves could single out an individual by the products they carry.

Suppose that contents were secured, perhaps by replacing explicit product information with a database pointer. This still leaves open the possibility that individuals might be physically tracked by the tags they carry. If tags respond predictably with the same pointer on every query, the holder of the tags may be tracked by readers which Mallory has installed

throughout an area. This violates the concept of "location privacy" [9]. Individuals should not have their movement tracked automatically. Similar issues arise in other pervasive computing systems, as well as Bluetooth networks [51].

Is location privacy such a serious risk? We are regularly filmed by CCTV cameras which increasing are coupled with facial recognition software. Anyone could follow you through public areas or hire a private investigator to do it for them. The difference with RFID technology is that tracking may be done automatically and with greater accuracy. Although most people may not care if they are tracked in public, groups like AIDS patients, religious worshipers, even sex shop purveyors need to be protected against being automatically singled out.

Suppose all unique identifying data is removed from tags at purchase time. For example, serial numbers are erased, but product and manufacturer codes are left intact. Customers could take advantage of product information without being tracked by a unique ID. Unfortunately, Mallory could still expose anyone carrying "embarrassing" products. Secondly, combinations of specific brands by be tracked collectively as "constellations". A distinct taste in a few brand names could act as a de facto identity.

Of course, tags could be completely disabled at checkout. However, it is plausible that all sorts of innovative home RFID applications might emerge. These might include "smart" medicine cabinets, pantries or refrigerators. There is also a large market for RFID in the waste disposal and recycling industries.

Insecure tags threats are not limited to individual privacy violations. Suppose a retail store installs a smart shelf system and stocks RFID-tagged products, similar to the scenario described in Section 1.3. In such a setting, the physical attacks available to Phyllis are not much use. We can assume the store will have security guards or video cameras to detect attacks against tag hardware.

However, Mallory could attack an insecure RFID system in a variety of ways. If tags lack read access control, Mallory could automatically query the entire store's inventory. By conducting periodic scans, Mallory could derive sales data; quite lucrative information should Mallory choose to offer her services as a corporate spy.

Alternatively, Mallory could re-write the contents of expensive items with data from

cheaper products. Similar attacks may be conducted against barcodes. One web site [77] even contained a database of barcode stickers for users to print out (although it was shut down within a week).

While a perceptive clerk might notice a fake sticker on the outside of a box, Mallory could wirelessly re-write an insecure tag. If she re-wrote RFID tags then came back later without any incriminating writing devices, Mallory could plausibly claim that she had nothing to do with the erroneous RFID tags. On the flip side, luxury items have begun to be labeled with RFID devices [79]. Mallory could help construct forgeries by writing valid data from a real luxury item onto a cheap knock-off.

Besides fraud, Mallory could use her skills to facilitate theft. Suppose the store had an automated checkout system - shoppers could bag their own items and be billed as they exited the store. Smart shelves would track when items were removed from shelves. If an inconsistency appeared, such as a removed item never exiting the store, security could be alerted.

Since Mallory has the ability to forge tags, she could defeat an automated checkout system. Mallory could remove an item from a shelf and deposit it into a metal-lined bag. Such products are available commercially [68] and may even include complimentary potato chips. Normally, a vanishing product would register as an anomaly. However, Mallory can replace the original product with a decoy device mimicking the original RFID label. The shelf will think that the item has been replaced, allowing Mallory to walk out the door. Mallory can even build a single device which will mimic many tags at once.

This *decoy attack* also works against RFID-based toll booths like E-ZPass [32], or subway turnstile systems. Mallory has an advantage because her decoy devices are not subject to the same physical and cost restrictions that RFID systems are. Whereas a store's tags must be cheap and easily incorporated into packaging, Mallory could build active, bulky devices. There is an incentive for theft as long as a decoy device costs less than the stolen merchandise.

Although Eve is weaker than Mallory, she may still represent a significant threat to security and privacy. Eve cannot arbitrarily query consumers' tags, but could eavesdrop next to a legitimate reader. For example, Eve could wait outside a pharmacy, next to a subway

turnstile or anywhere tags may be queried by an authorized reader. Eve may conduct industrial espionage as well. By recording a store's own inventory queries, Eve could extract the same information as Mallory's active attacks.

Even Tracy poses a threat to security and privacy. Although she cannot query or eavesdrop, Tracy may detect the presence of tags and queries. Individuals could be tracked if they consistently carry a specific number of devices, especially if they carry an unusually high number. Tracy might even be able to glean inventory data. Knowing that a reader queried x tags in the produce section or that a milk truck arrived carrying y tags is still valuable, albeit imprecise, information.

The weakest attacker, Denise, also poses a threat. Denise cannot derive useful information from an RFID system, but can launch denial of service attacks against the system. She could flood RF channels with noise to disrupt or garble communication. Denise might even be able to conduct a low-level directed energy attack [94] to destroy tags. Analogously, someone could easily destroy barcodes by tearing them off or writing over them. However, Denise may be able to automatically disrupt RFID systems on a widespread scale via RF.

Chapter 4

Practical Security Assumptions

In order to propose specific security mechanisms, we will state several assumptions regarding the cost, resource limitations and performance requirements in a low-cost RFID system. We define a "baseline tag" with a cost of US$0.05 (5¢) per tag, not including reader or back-end database expenses. However, we may assume that readers and back-end systems have ample storage, computation and communication resources. Our baseline 5¢ tag will most certainly be passive and will be assumed to have 96 bits of storage. It will be required to support 100 read operations per second and to operate in an environment densely populated with tags.

The clock cycles available during a tag read operation depend on the operating frequency, tag technology and on various other factors. For example, tags operating at 915 MHz are required to hop frequencies every 400 ms due to RF regulations. The clock on a tag may operate at some multiple or fraction of the communication frequency. We assume our baseline tag will have 10,000 tag clock cycles for security computation. This is a somewhat arbitrary assumption, but is a useful ceiling when considering practical security mechanisms.

Power dissipation is another issue among tags and also depends on the tag technology, operating frequency, power coupling mechanism and various other factors. We do not make any assumption on the maximum power dissipation of a tag, although it could be an important factor limiting security related computations. Anecdotally, some technologies have a maximum power dissipation of 10μwatts [99].

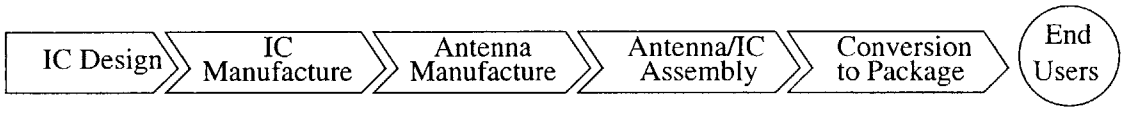

Figure 4-1: Five steps of tag manufacturing.

The five major steps in the manufacturing process of a RFID tag are IC design, IC manufacture, antenna manufacture, antenna/IC assembly and conversion to packaging. This process is depicted in Figure 4-1. IC design is considered a fixed cost and is not factored into our per tag costs. In 2003, a large portion of the per tag cost must be allocated to antenna manufacture, antenna/IC assembly and conversion to packaging. These steps cost approximately $1¢$ each, leaving roughly 1-$2¢$ for IC manufacturing.

Raw silicon is the major cost barrier in IC manufacturing. The cost per-mm^2 of silicon is roughly $4¢$ [89]. This cost has remained more or less stable for many years. At this price, the IC manufacturing budget dictated for the baseline tag only allows for a chip of about a .25mm^2 in size. The number of gates per mm^2 (gate density) depends on the line width of the IC manufacturing technology used. Table 4.1 shows typical gate densities and manufacturing costs. Current RFID technologies will likely have .5μm or .35μm line widths. We will make a liberal assumption that there will be 200-2000 gates allocated to security for our baseline tag. As a rule of thumb, each additional 1000 gates will cost $1¢$.

These estimated gate counts are far below what is necessary for standard public-key and symmetric encryption algorithms. In fact, just the private key for most public-key algorithms dwarfs the tag's entire storage, even schemes with relatively small keys such as NTRU [46, 72] and elliptic curve cryptosystems [57]. Symmetric algorithms fare no better. Hardware implementations of DES [37] and AES [39] run on the order of 20,000-30,000 gates [17]. This exceeds the resources available for the *entire* RFID design. Implementations of standard cryptographic hash functions, like SHA-1 [38] also cost roughly 20,000 gates. Even the aptly named Tiny Encryption Algorithm [109, 110] is too costly for today's tags, although could be feasible in the near future.

As discussed in Chapter 3, tag memories are assumed to be susceptible to physical attacks (in the parlance of Chapter 3, Phyllis' attacks). Adding tamper resistance or physical

Line Width	Gates/mm^2	Fab Cost ¢/mm^2
.8μm	1,500	2.5
.5μm	4,000	3
.35μm	10,000	4
.25μm	38,000	6
.18μm	60,000	8

Table 4.1: The gate density and foundry costs of several different line widths.

shielding is prohibitively expensive [107]. Due to these concerns, tags left in isolation cannot be trusted to securely store long-term shared secrets.

A distinct issue arising in implementations of passive tags is the *asymmetric channel strength* between the forward (tag-to-reader) and backward (reader-to-tag) channel. Since passive tags receive power via the forward channel, it is much stronger than the backward channel. As a result the forward channel may be monitored from a much greater distance than the backward channel. For example, a 915 MHz passive tag may have a 3-meter operating range, yet its forward channel may be monitored from 100 meters. In ideal conditions, a 915 MHz forward channel could theoretically be monitored from a kilometer. This relation is illustrated in Figure 4-2.

This asymmetry in channel strength could lead to eavesdropping, that is, Eve's attacks. Generally, it will be assumed that only the forward channel may be monitored without detection. To monitor the backward channel, an eavesdropper would have to be within the short range of the backward channel (e.g. 3 meters). The threat of backward-channel eavesdropping should not be discounted completely. Eve could still scatter listening devices or could attempt to piggy-back a bugging device on a legitimate reader. However, these attacks are more costly and easier to detect than forward-channel eavesdropping.

The Binary Tree-Walking anti-collision algorithm discussed in Section 2.2.4 suffers a vulnerability due to the asymmetry of channel strengths. Recall that a reader conducting the tree-walking protocol will broadcast each ID bit of the tag it is singulating. Because this is over the forward channel, an eavesdropper may monitor the entire ID from a safe distance. Since every tag must be singulated, a long-range eavesdropper could obtain the ID of every tag which is read. Proposals to address this issue are presented in Section 5.4.

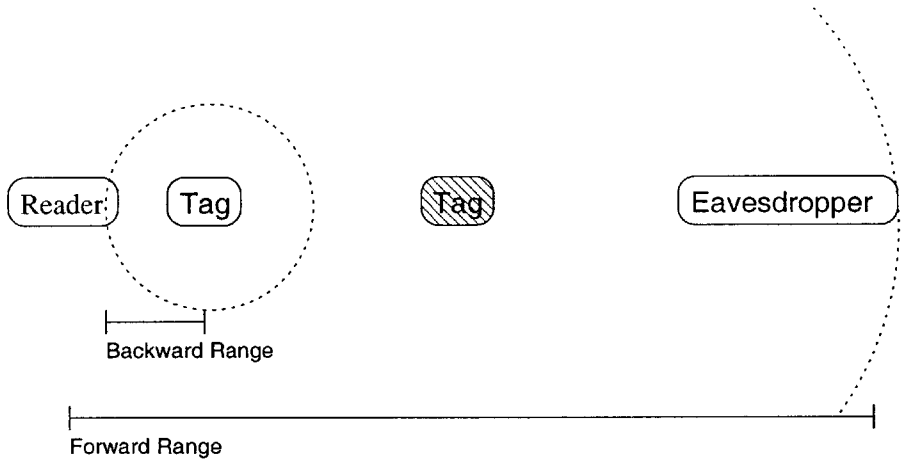

Figure 4-2: The reader will detect the nearby tag, but cannot detect the shaded tag. A distant eavesdropper may monitor the forward channel, but not the tag responses.

Tags may be equipped with a physical contact channel, as found on smart cards. This channel may be used for critical functions or for *imprinting* [97]. Imprinting is the process of setting the ownership of an uninitialized device. Tag imprinting will likely be a time-consuming process which requires physical possession of a tag. Additionally, we may assume that tags contain some optical information such as a barcode or human readable digits. Juels and Pappu use this type of printed data to corroborate tag data in a proposal to secure RFID-labeled Euro notes [52].

Tags will be assumed to have a "ping" mechanism to reveal their presence. This functionality is akin to the Class 0 electronic article surveillance (EAS) tags discussed in Section 2.1.1. Anyone may ping a tag, which responds with some non-identifying signal. Tags will also be equipped with a "kill" command. Killing a tag will be a slow, physical process, similar to imprinting, that renders a tag permanently inoperable. The act of killing a tag might involve disconnecting the antenna, shorting a fuse or subjecting a tag to high-energy microwaves.

In summary, our baseline tag design is specified in Figure 4-3. The proposals presented in Chapter 5 will largely be designed for this specification.

- **Class 1 EPC Tag:** Passively powered with 96 bits read-only memory.
- **Range:** 3m operating, 100m forward channel, 3m backward channel.
- **Anti-Collision:** Either deterministic or probabilistic algorithm.
- **Performance:** 100 read operations per second.
- **Clock Cycles per Read:** 10,000 clock cycles
- **Security Gate Count:** 200-2000 gates
- **Logical Operations:** *read, ping*
- **Physical Operations:** *imprint, kill*

Figure 4-3: Example specification for a low-cost RFID tag.

Chapter 5

Security Proposals

The issues presented in Chapter 3 must be addressed while working under the constraints specified in Chapter 4. We made the assumption that tags are vulnerable to physical attacks against hardware. As a result, our primary concerns are active attacks and eavesdropping attacks. These attacks may violate individual privacy as well as leak sensitive inventory data. Traffic analysis attacks also present a threat, particular to an individual's location privacy and to organizational logistics data. Denial of service may also be a potentially expensive and disruptive attack.

Active querying attacks may be addressed by limiting who is permitted to read tag data through access control. Eavesdroppers may be dealt with by ensuring that tag contents are not broadcast in the clear over the forward channel. In Section 5.1 we present a low-cost access control mechanism based on one-way hash functions. This mechanism is referred to as a *hash lock*. Section 5.2 presents a randomized version of hash locks which prevents tag tracking. Section 5.3 defines requisite hash properties and explores various approaches to building low-cost hash functions appropriate for hash locks. Section 5.4 offers two variants of the binary tree-walking anti-collision algorithm offering greater security against long-range eavesdroppers. Section 5.5 contains several simple concepts which may strengthen security, particularly in detecting and preventing denial of service attacks.

5.1 Hash Lock

Access control mechanisms are frequently based on public-key cryptographic primitives or symmetric primitives requiring secure key distribution. Current RFID tags lack the computational resources to support these traditional access control mechanisms. Hash locks are a simple access control mechanism based on one-way hash functions. A definition of one-way hash functions appears in in Section 5.3.1. The hash lock scheme also appears in [91] and [108] by the author.

Tags in the hash lock scheme will each be equipped with a hash function. In practice, a hardware-optimized cryptographic hash would suffice. Candidate designs for such low-cost hashes are presented in Section 5.3. Each hash-enabled tag in this design will have a portion of memory reserved for a temporary *metaID*. Tags will operate in either a locked or unlocked state. These states may be arbitrarily defined for different tag designs. We will assume that an unlocked tag offers its complete functionality to any nearby reader.

A tag owner locks tags by first selecting a key at random, then computing the hash value of the key. The hash output is designated as the *metaID*, that is $metaID \leftarrow hash(key)$. The tag owner will then store the metaID on the tag and toggle it into a locked state. Writing the metaID may occur either over the RF interface or over a physical contact channel for added security. Upon receipt of a metaID value, the tag enters its locked state. While locked, a tag responds to all queries with only its metaID and offers no other functionality. Finally, the tag owner will store the key and metaID in a back-end database, indexed on the metaID. This protocol is summarized in Figure 5-1.

1. Reader R selects a random key and computes $metaID := hash(key)$.
2. R writes $metaID$ to Tag T.
3. T enters the locked state.
4. R stores the pair $(metaID, key)$ locally.

Figure 5-1: Protocol for locking a hash lock.

To unlock a tag, the owner first queries the metaID from the tag and uses this value to look up the key in a back-end database. The owner transmits this key value to the tag, which hashes the received value and compares it to the stored metaID. If the values match, that is $hash(key) == metaID$, then the tag unlocks itself and offers its full functionality to any nearby readers. This protocol is summarized and illustrated in Figures 5-2 and 5-3. To prevent hijacking of unlocked tags, they should only be unlocked briefly to perform a function before being locked again.

1. Reader R queries Tag T for its $metaID$.
2. R looks up $(metaID, key)$ locally.
3. R sends key to T.
4. If $(hash(key) == metaID)$, T unlocks itself.

Figure 5-2: Protocol for unlocking a hash lock.

Based on the difficulty of inverting a one-way hash function, this scheme prevents unauthorized readers from reading tag contents. Spoofing attempts may be detected under this scheme, although not prevented. An adversary may query a tag for its metaID, then later spoof that tag to a legitimate reader in a replay attack. A legitimate reader will reveal the key to the spoofed tag. However, the reader may check the contents of the tag (often collectively referred to as a tag's ID) against the back-end database to verify that it is associated with the proper metaID. Detecting an inconsistency at least alerts a reader that a spoofing attack may have occurred.

The hash lock scheme only requires implementing a hash function on the tag and managing keys on the back-end. This may be economical in the near future, possibly by using suggestions from Section 5.3. Hash locks can be extended to provide access control for multiple users or to other tag functionality, such as write access. Tags may still function as object identifiers while in the locked state by using the metaID for database lookups. This allows third-party users to build their own databases and to take advantage of tag functionality without necessarily owning the tags. Unfortunately, since the metaID acts as an identifier, tracking of individuals is possible under this scheme.

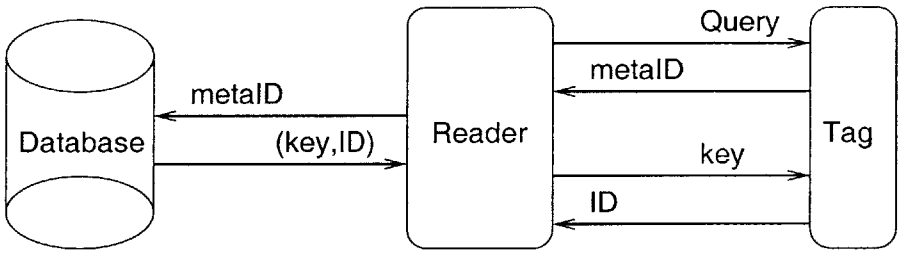

Figure 5-3: A reader unlocks a hash-locked tag.

5.2 Randomized Hash Lock

Preventing tag tracking motivates an additional mode of operation. While in this mode, a tag must not respond predictably to queries by unauthorized users, but must still be identifiable by legitimate readers. We present a practical heuristic based on one-way hash functions, best suited for consumers with a small number of tags. We also offer a theoretically stronger variant based on pseudo-random functions (PRFs). This problem in general, as well as other related problems are discussed further in Section 7.1.

As in Section 5.1, tags are equipped with a one-way hash function, but now also have a random number generator. We assume that a legitimate tag reader will "know what she owns" before scanning tags. An unlocked tag may be locked with a simple instruction from a reader; no protocol is necessary. To unlock a tag, a reader first sends a simple query. Tags respond to this query by generating a nonce R chosen uniformly at random. The tag then hashes this nonce concatenated with the tag ID. Finally, the tag sends a reply to the reader consisting of both the nonce and the hash output, that is with the pair $(R, h(ID||R))$.

When a legitimate reader receives the pair $(R, h(ID||R))$, it performs a brute-force search of all its known IDs by hashing each of them concatenated with R until it finds a match. Recall that readers are assumed to "know what they own". Once the reader finds a match, it can unlock the tag by sending the ID value. Alternatively, since the reader now knows the ID value, it may leave the tag locked. This protocol is summarized in Figure 5-4 and illustrated in Figure 5-5.

In practice, FPGA-based SHA-1 cores offer a data throughput of approximately 400-600 Mbps on 512 bit blocks which is approximately 750,000-1,000,000 hash operations

1. Reader R queries Tag T.
2. T generates a random nonce R and computes $hash(ID||R)$.
3. T sends $(R, hash(ID||R))$ to R.
4. R computes $hash(ID_i||R)$ for all its known ID_i values.
5. If R finds a match such that $hash(ID_j||R) == hash(ID||R)$, R sends ID_j to T.
6. T unlocks itself if it receives $ID_j == ID$.

Figure 5-4: The Randomized Hash Lock Disable Protocol

per second [2]. ASIC implementations would offer even better performance. In 1996, assembly code versions of SHA-1, MD5 and RIPEMD on a Pentium ran at rates of 48.7, 113 and 82 Mbps, respectively [14]. Anecdotally, in 2003 a 1.6 Ghz Pentium can perform about 460,000 SHA-1 hash operations per second. An ideal hash function would be one which could be implemented to take few gates and many cycles on a tag, but could be implemented to take few cycles on a reader. This is discussed further in Section 5.3.

This scheme is impractical for owners of huge numbers of tags who require read rates of 100-200 tags per second. However, it may be more feasible for holders of a relatively small number of tags. Since location privacy is less of a concern for retail stores than for individuals, retailers might employ a regular hash lock, then engage the randomized version for a consumer upon purchase.

One issue is how "legitimate" readers come to know about their tags. When a product is sold, the value of its ID must be transfered with it. Otherwise, the new owner cannot read the tag. Section 5.5.6 discusses one mechanism which would allow a new owner to access their tags.

Although this scheme may suffice in practice, but is not theoretically robust. The formal definition of a one-way function only establishes the difficulty of inverting the function output. There is no provision of secrecy, technically allowing bits of the input to be revealed. This is fully explained in Section 5.3.1. We can use a stronger primitive to ensure ID bits are not leaked.

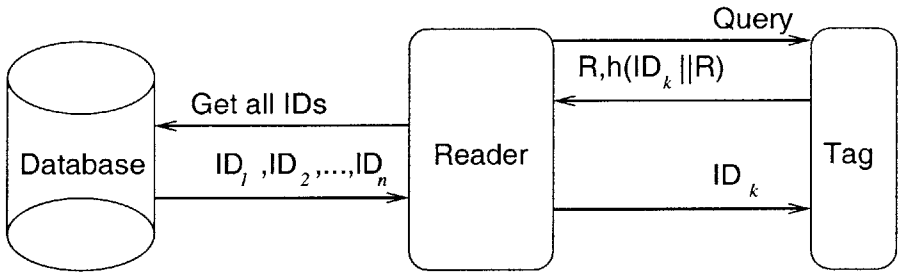

Figure 5-5: A reader unlocks a tag whose ID is k in the randomized hash lock scheme.

To address this issue, suppose each tag shares a unique secret key k with the reader and supports a pseudo-random function ensemble, $\mathcal{F} = \{f_n\}_{n\in\mathbb{N}}$. When queried, tags will generate a random value, R, and reply with $(R, ID \oplus f_k(R))$. The reader will once again perform a brute-force search, using all its known ID/key pairs to search for a match.

A minor fix allows readers to only store tag keys on the back-end, without needing to also store the tag IDs. Tags may pad their ID with its hash, then reply with $(R, (ID||h(ID))\oplus f_k(R))$. Readers may identify tags by computing $f_k(R)$ for all their known keys, XORing it with the second part of the tag's response, and searching for a value ending in the form $(x||h(x))$. To anyone without the key value, the tag's output is random and meaningless.

It is unknown whether PRF ensembles may be implemented with significantly fewer resources than symmetric encryption. There may be no be practical difference in the context of low-cost RFID tags. Many symmetric encryption algorithms employ PRFs as a core building block in a Luby-Rackoff style design [63]. The minimal hardware complexity of a PRF ensemble remains an open problem [61].

5.3 Low-Cost Hash Functions

The proposals in Sections 5.1 and 5.2 rely on low-cost hash functions as a fundamental building block. Typical commercial implementations of standard hash functions such as SHA-1 take on the order of 20,000-30,000 gates [2, 17]. This cost exceeds the resources available to an entire low-cost RFID design.

As discussed in Chapter 4, we assume low-cost tags will be required to perform 100-

200 read operations per second and will have 200-2000 gates available for security. Many commercial implementations of hash functions are optimized for speed, rather than gate count. RFID systems allow much greater flexibility in this in respect. Perhaps 10,000 clock cycles may be available for security functions. The relative abundance of clock cycles suggest using *few gates, many cycles* as a design principle when considering low-cost hash designs.

A comprehensive analysis of cryptographic hash functions is available in Preneel's Ph.D. thesis [74]. Bakhtiari, Safavi-Naini and Pieprzyk present a general survey on cryptographic hash functions [5]. In Section 5.3.1 we will define one-way and collision-resistant hash functions. Section 5.3.2 presents several historical design approaches, which are not appropriate for low-cost RFID. Sections 5.3.3 and 5.3.4 suggest two candidate design paradigms for low-cost RFID functions.

5.3.1 Hash Definition

The definitions presented in this section are largely based on Menezes, van Oorschot and Vanstones' book [64]. At minimum, a hash function h is an efficiently computable function which maps an arbitrary length input to a fixed length output; i.e. $h : \{0,1\}^* \rightarrow \{0,1\}^n$. Many trivial functions fall under this loose definition, so it is of little use. The following three properties prove to be more useful:

- *preimage resistance* - For all outputs y, it is computationally infeasible to find any input x such that $h(x) = y$ given no corresponding input is known.
- *2nd-preimage resistance* - Given x, it is computationally infeasible to find $x' \neq x$ such that $h(x) = h(x')$.
- *collision resistance* - It is computationally infeasible to find any pair of inputs x and x' such that $h(x) = h(x')$. Note the freedom of choice on both inputs.

A *one-way hash function* (OWHF) is a hash function which offers preimage and 2nd-preimage resistance. This may be thought of simply as being "difficult to invert". A *colli-*

sion resistant hash function (CRHF) is a hash function which is 2nd-preimage resistant and collision resistant. Although not necessary, most CHRFs are also one-way in practice.

Although difficult to invert, one-way hash functions do not necessarily "hide" information. For example, suppose we are given a one-way hash function h'. Define a second hash function $h(x||y) = h'(x)||y$. The output of h is difficult to invert based on the hardness of inverting h'. However, the second half clearly leaks information about the input. In the context of RFID, recall the randomized hash lock scheme in Section 5.2. Under this scheme, the pair $(R, h(R||ID))$ was sent by a tag. If h were defined as above, this value would be $(R, h'(R)||ID)$. This leakage of information clearly defeats the purpose of using a hash function in the first place. In theory, a randomized hash lock should rely on a pseudo-random function or a perfect one-way function [16]. Despite these theoretical insecurities, heuristic hash functions sufficiently hide information in practice.

5.3.2 Design Approaches

Several varied approaches to building hash functions have been employed in practice. Many of these approaches are prohibitively expensive for low-cost tags. Two main classes of hashes are those with theoretical proofs of hardness, such as hashes based on modular arithmetic or NP-Completeness, and heuristic hashes used in practice. This division is analogous to the differences between public-key and symmetric cryptosystems. In fact, there are examples of each class of hashes which rely on the same underlying public-key and symmetric primitives.

The theoretical based hashes include those based on modular arithmetic, algebraic matrices and "hard" problems like the Knapsack problem. Modular arithmetic based hashes rely on the underlying hardness of factorization or finding discrete logarithms in Galois fields. The RSA [86] and El Gamal [36] cryptosystems are based on the same underlying problems respectively. The digest size of modular math based hashes depends on the size of the modulus. Sufficiently large moduli will be much larger that the space available in an RFID tag. Modular arithmetic operations are also too computationally intensive and would require too many gates to implement on low-cost tags. Relying on underlying hardness of

rings, lattices or elliptic curves [46, 57] does not offer much promise for tags in the near future either.

Theoretically secure hashes may also be based on algebraic matrices. For example, given an $n \times n$ secret matrix K, the hash of message M may be defined as $H(M) = M^t K M$. Several weaknesses to these types of systems are pointed out in [74]. Unfortunately, hashing a message of, for example, 128 bits would require a key matrix of size approximately 512 bytes. Matrix operations of this size are beyond what is feasible for low-cost tags. However, a small matrix-based S-box or sub-hash function might be a useful building block in hash function design.

A third class of hashes rely on the hardness of the Knapsack Problem. The Knapsack Problem is stated as follows: Given a set of integers $S = S_1, \ldots, S_k$ and an integer n, find some subset $T \subseteq S$ such that $\sum_{t_i \in T} t_i = n$. Merkle and Hellman initially used this problem in public-key cryptosystem [65]. Harari [45], Damgard [29] and Zémor [114] later proposed both additive and multiplicative Knapsack-based hashes. Damgard's scheme was broken by Camion and Patarin [15]. Preneel points out weaknesses in Harari's scheme [74]. The heavy computational and storage requirements of Knapsack based hashes are beyond low-cost tag resources as well.

Heuristic hashes may be dedicated functions or may rely on block ciphers as a fundamental building block. Of the latter, many schemes have been proposed. Examples include N-Hash [10], the ISO's "data integrity mechanism" [48], and Lai, Rueppel and Wooliven's "cryptographic checksum" [62]. Preneel, Govaerts and Vandewalle [75] examined 64 basic ways to assemble a hash function from a block cipher. They showed attacks against 52 out of 64. Black, Rogaway and Shrimpton later showed the security of the remaining 12 through black-box analysis [12]. Since we are assuming that block ciphers will be too expensive to implement, this currently is not a feasible option. However, it would be an efficient use of resources if future designs could use a block cipher for both hashes and encryption.

Typically, dedicated functions designed specifically to be hashes are used in practice. The MD-family of hash functions includes MD2 [55], MD4 [81, 82] and MD5 [83]. Serious flaws have been found in MD2 [87]. Dobbertin showed how collisions in MD4 could

be found within few minutes on regular PC [30]. MD5 is generally regarded as secure, although Dobbertin's technique may be extended to find collisions in the compression function of MD5. This attack does not find collisions in MD5, but may be an important first step.

SHA-1 [38] is another commonly used dedicated hash function, as well as HAVAL [115] and RIPEMD [31]. Generally, these algorithms have been optimized for speed and easy software implementation. Most implementation costs dwarf the resources available in RFID. We present two design approaches which may be appropriate for low-cost systems in Sections 5.3.3 and 5.3.4.

5.3.3 Cellular Automata

Cellular automata (CA) are finite state machines whose state transitions depend solely on nearby neighbors. Wolfram offers a treatise on the subject in [113]. The simplest cellular automata system are binary and one-dimensional. Each cell's next state depends on its own state and those of its immediate neighbors. So cell i's value in time step $t + 1$, i.e. $c_{i,t+1}$, depends on $c_{i-1,t}, c_{i,t}, c_{i+1,t}$. Since each of the eight possible states has two possible outputs, there are 256 total binary one-dimensional CA systems.

Many of these systems behave predictably. However, several exhibit "random" or "chaotic" properties. Wolfram explored using these properties in cryptography and random number generation in [111] and [112]. In particular, Wolfram analyzed a particular cell of CA Rule #30, defined as $c_{i,t+1} = c_{i-1,t} \oplus (c_{i,t} \vee c_{i+1,t})$, implemented in a cyclic register.

A register of size n requires roughly $2n$ logic gates to implement a CA. Figure 5-6 shows a diagram of a register implementing Rule #30 on a single cell. Figure 5-7 portrays the time evolution of Rule #30 running in a fixed-sized cyclic register for several hundred iterations. Initially, the register contained a single "on" cell.

An early implementation of a CA-based hash was presented by Damgard [29]. Daemen, Govaerts and Vandewalle showed an insecurity in Damgard's scheme. They propose their own CA-based hash called Cellhash [27], as well as an improved version called Sub-

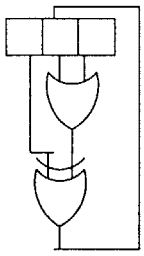

Figure 5-6: Implementation of a CA on a single cell of a fixed-sized cyclic register.

Hash [28]. While no breaks in Cellhash have been published, Preneel points out a few disadvantages [74]. Regardless, Cellhash and other CA-based hashes may be an appropriate paradigm for low-cost RFID tags. CA hashes scale well as the size of the hash digest increases. Another benefit is that tags may already have a register used for anti-collision which may also be used for hashing.

One drawback of CA hashes on RFID tags are that many parallel calculations may consume too much power. Fortunately a CA may be arbitrarily serialized at the expense of performance. A CA may even be built out of a feedback shift register and a single pair of gates. More complicated feedback shift register based hash functions are discussed in Section 5.3.4.

A second problem with CA based hashes are that they necessarily require a large number of clock cycles. There is no clear way to reduce cycles. In fact, Wolfram has speculated that CA are computationally irreducible – their output may only be found by actually stepping through each calculation [113]. A reader implementing a randomized hash lock using a CA hash would have to run through many cycles for every one of their known tags. Of course, this may be an inherent weakness of randomized hash locks. Other potential approaches are discussed in Section 7.1.

5.3.4 Non-Linear Feedback Shift Registers

A feedback shift register consists of a shift register and a feedback function. Specific bits in the shift register are "tapped" and fed into the feedback function. The register is shifted each time an output bit is needed. The feedback function will output a bit which is input

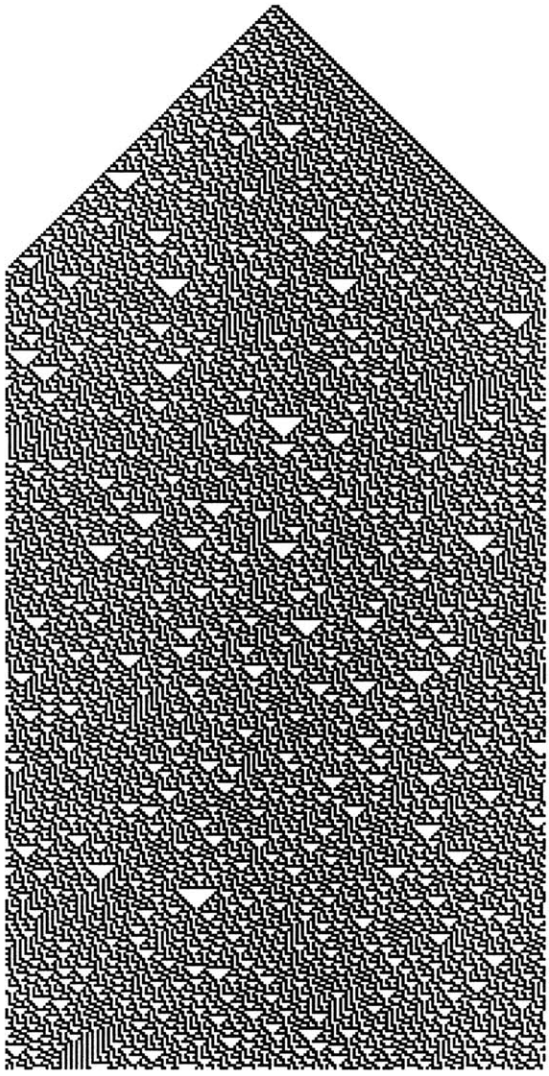

Figure 5-7: Example of the "noisy" output from a cellular automata system.

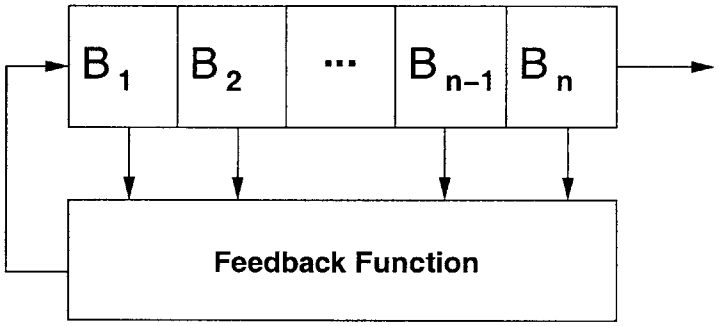

Figure 5-8: A feedback shift register.

back into the register. This is illustrated in Figure 5-8.

The feedback function may be linear or non-linear. The resulting structures are referred to as Linear Feedback Shift Registers (LFSR) and Non-Linear Feedback Shift Registers (NLFSR). Ernst Selmer [95] conducted early work on the mathematics behind LFSRs. Soloman Golomb, an NSA cryptographer, incorporated Selmer's results with his own in his seminal book [44].

More complicated feedback functions may be employed. For example, multiple bits may be replaced by a *substitution box* or (S-Box), or a *permutation box* (P-Box). Similar to cellular automata, a low-cost hash function will rely on a simple feedback function iterated many times.

Two useful concepts in a heuristic hash design were introduced by Claude Shannon over 50 years ago [96]. These concepts are *confusion* and *diffusion*. Confusion is the property that the statistical relationship between the hash input and output should be too complicated for adversaries to exploit. Diffusion is the property that the influence of a single bit of input should be spread among many bits of output. In other words, predictable patterns of input should be hidden in the output.

Putting this in the context of a NLFSR-based hash design, the hope is that a complicated feedback function should create confusion. Iterating this function over many cycles and properly positioning the tap points will help diffuse the influence of input bits. We offer a design approach consisting of a small P-box, with tap points at powers of two. This P-box will operate on a working register that is twice the size of the eventual hash output.

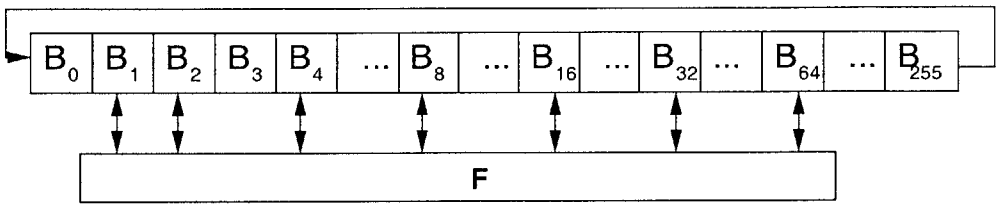

Figure 5-9: A proposed NLFSR hash architecture.

After iterating many cycles, half the working register will be discarded. The remaining half becomes the hash function output. We illustrate this design in Figure 5-9.

Figure 5-9 portrays a NLFSR-based hash with a digest size of 128 bits. The function F is a 7-bit permutation. In a single iteration, bits 1, 2, 4, 8, 16, 32 and 64 are tapped and input into F. The resulting output is written back to the same bits. Finally the register is shifted a bit to the right. After a large number of iterations, the second half of the register is truncated. The remainder is the hash output.

Before truncation, the hash function is a permutation. Inverting it would entail simply running the function in reverse. This ensures that information is not "lost" during iterations, which would limit the range of the final result. Minimally, the two tap points at Bit 1 and 2 ensure that every input bit can influence every output bit. The additional tap points are intended to diffuse the influence of each bit more quickly.

The selection of 7 tap points is solely due to resource limitations. A 7-to-7 permutation requires a lookup table of size 128×7, which is approximately 900 bits. A standardized permutation could be hardwired into tags relatively cheaply. Assuming each iteration takes two cycles, this function could be iterated about 5,000 times. A 256 bit register will be shifted through completely about 20 times. This should rapidly diffuse each input bit evenly over the output.

This is an example of one possible low-cost design. The function F, the tap points and the register size may all be varied. Perhaps the only assumption which may not be realistic is that tags will have a 256-bit shift register. Today's tags only have 96-bits of read only memory. However, there are already standards being developed for 128-bit and 256-bit EPC tags [67].

As with CA hashes, a potential problem with NLFSR based randomized hash locks is that readers would need to compute hash outputs for every known tag. Unlike CA systems, NLFSRs might be parallelized on the reader side to offer greater performance. Our assumption is that readers will have ample resources in comparison to tags. A parallelized NLFSR essentially trades off greater gate counts for faster performance.

5.4 Secure Anti-Collision

The Binary Tree-Walking anti-collision algorithm discussed in Section 2.2.4 has an inherent security flaw due to the asymmetry between forward and backward channel strengths. Every bit of every singulated tag is broadcast by the reader on the forward channel. At certain operating frequencies, a long-range eavesdropper could monitor these transmissions from a range of up to 100 meters and recover the contents of every tag. Sections 5.4.1 and 5.4.2 present two secure variants of the normal tree-walking scheme.

5.4.1 Blinded Tree-Walking

One security concern is the strong signal of the reader-to-tag forward channel discussed in Chapter 4. Depending on the tag operating frequency, eavesdroppers could monitor this channel from hundreds of meters and possibly derive tag contents. Of particular concern is the binary tree-walking anti-collision algorithm specified in Section 2.2.4, because the reader broadcasts each bit of the singulated tag's ID over the "loud" forward channel.

We present a variant of binary tree-walking which does not broadcast insecure tag IDs on the forward channel and does not adversely affect performance. This scheme originally appeared in [108] under the name "Silent Tree-Walking". Assume a population of tags share some common ID prefix, such as a product code or manufacturer ID. To singulate tags, the reader requests all tags to broadcast their next bit. If there is no collision, then all tags share the same value in that bit.

A long-range eavesdropper can only monitor the forward channel and will not hear the tag response. Thus, the reader and the tags effectively share a secret bit value. When a collision does occur, the reader needs to specify which portion of the tag population should

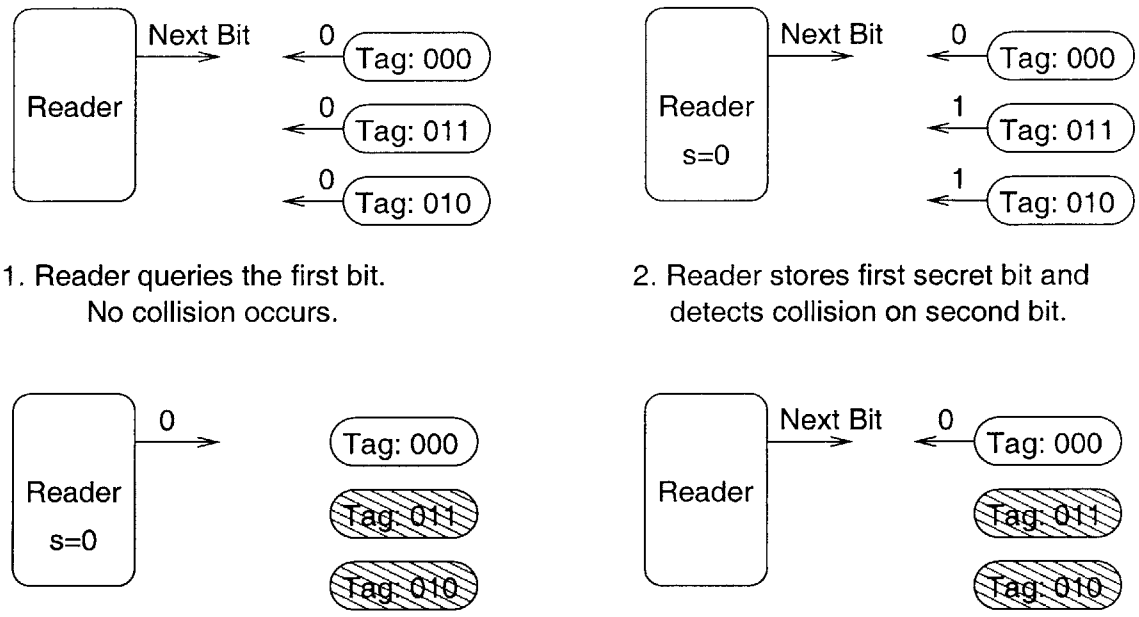

Figure 5-10: A reader singulates Tag 000 with a blinded tree-walking algorithm.

proceed. If no collisions occur, the reader may simply ask for the next bit, since all tags share the same value for the previous bit.

Since we assumed the tags shared some common prefix, the reader may obtain it as a shared secret on the backward channel. The shared secret prefix may be used to conceal the value of the unique portion of the IDs. Suppose we have two tags with ID values $b_1 b_2$ and $b_1 \overline{b_2}$. The reader will receive b_1 from both tags without a collision, then will detect a collision on the next bit. Since b_1 is secret from long-range eavesdroppers, the reader may send either $b_1 \oplus b_2$ or $b_1 \oplus \overline{b_2}$ to singulate the desired tag without revealing either bit. Figure 5-10 illustrates a reader performing blinded tree-walking on two bits.

Eavesdroppers within the range of the backward channel will obviously obtain the entire ID. However, this blinded tree-walking scheme does effectively protect against long-range eavesdropping of the forward channel with little added complexity. Performance is identical to regular tree-walking, since a tag will be singulated when it has broadcast its entire ID on the backward channel.

5.4.2 Randomized Tree-Walking

The Blinded Tree-Walking algorithm in Section 5.4.1 may be simplified with randomization. The general idea behind Randomized Tree-Walking, due to Rivest [85], is for each tag to generate a temporary random *pseudo-ID* each tree traversal. Related ideas appear in [4]. The reader will perform a normal tree-walking scheme on the pseudo-ID values. Once a tag is singulated, it will send its normal ID over the backward channel.

In this scheme, mostly pseudo-ID bits will be broadcast over the forward channel. No assumption about common tag prefixes or secret key management on the reader side is necessary. Randomization does incur a performance penalty due to additional communication costs, which is discussed later in this section. Performance may be traded off for a slight privacy leakage.

In order to ensure that no tags are overlooked, the reader must maintain power to all nearby tags until it has read them all. Essentially, tags will have a new pseudo-ID each tree traversal session. Once power is cut, all tags will forget their pseudo-ID.

Pseudo-ID bits may actually be generated on the fly. The reader will query all tags for a random bit. If a collision is detected, it will direct all those with a particular bit to sleep. Tags must keep track of the bit position they were put to sleep in. After several bits without collision, the reader will be convinced that a tag has been singulated. In the unlikely event that two tags generate the same random bits, the reader will still be able to detect collisions on the regular ID. The probability of this event may be made arbitrarily small at the cost of communication performance.

The reader's traversal algorithm is presented in Figure 5-11. The function "Traverse" receives two arguments: i is the current bit position and $count$ is the number of consecutive bits without collisions. If $count$ exceeds some a priori threshold, the reader will assume a tag has been singulated and will attempt to read its ID.

The main disadvantage of randomized tree-walking is the additional communication cost of the pseudo-ID. Recall that both regular and blinded tree-walking only broadcast the length of a tag's ID. Another minor issue is that tags will need a small bit of state to keep track of when they are suspended (i.e. the value i in Figure 5-11).

Traverse(i, $count$)
    b_i := Read random bit i from all active tags.
    if collision on b_i is detected:
        Suspend all tags with $b_i == 1$.
        Each suspended tag stores i.
        Traverse(i+1, 0).
        Wake up all tags suspended on bit i.
        Traverse(i+1, 0).
    else if no collision on b_i is detected:
        if ($count > threshold$) **Tree-Walk** remaining tags.
        else Traverse(i+1, count+1).

Figure 5-11: The Randomized Tree-Walking Algorithm

The choice of pseudo-ID length depends on the size of the nearby tag population. For a population of n tags, suppose m pseudo-ID bits are used. The number of tags randomly selecting a particular pseudo-ID will follow a Poisson distribution. If $\lambda = \frac{n}{2^m}$, the expected number of pseudo-IDs with k tags will be around $2^m e^{-\lambda}\frac{\lambda^k}{k!}$. Suppose $n = 2000$ and $m = 16$, then $\lambda = .03$. The expected number of pseudo-IDs with k tags is given in Table 5.1.

If 96-bit IDs are used, transmitting a 16-bit pseudo-ID represents about a 17% performance penalty. Note that if collisions among pseudo-IDs are dealt with by naive tree-walking, each collision of k tags on the same pseudo-ID value essentially leaks k bits of ID data. Silent Tree-Walking might be used to hide some of this data, although that requires the assumption of a common prefix. Alternatively, an adaptive scheme might only ask for pseudo-ID bits when real ID collisions occurred.

In the example given in Table 5.1, 2000 tags with 96-bit IDs contain approximately 192,000 bits of data. The collisions will leak approximately 30 bits of this data on each tree traversal. This may be acceptable in many applications, although over many tree traversals the leakage could add up. The choice of pseudo-ID length and how to handle pseudo-ID collisions may be left up to tag users depending on their particular privacy and performance requirements.

k	Pseudo-IDs with k Tags	Comment
0	63599	Most pseudo-IDs will not be selected.
1	1907	Most tags will generate a unique pseudo-ID.
2	28	A few pairs of tags will collide on the same pseudo-ID.
3	0.29	More than two tags will rarely collide.

Table 5.1: Expected distribution of 2000 tags over random 16-bit pseudo-IDs.

5.5 Other Proposals

5.5.1 Asymmetric Key Agreement

Readers may take advantage of the asymmetry of the forward and backward channels to transmit sensitive values, such as keys. Suppose a reader needs to transmit the value v to a singulated tag. That tag can generate a random value r as a one-time-pad and transmit it in the clear on the backward channel. The reader may now send $v \oplus r$ over the forward channel. If eavesdroppers are outside the backward channel, they will only hear $v \oplus r$, and v will be information theoretically secure.

5.5.2 Chafing and Winnowing

Another deterrent to forward channel eavesdropping is to broadcast "chaff" commands from the reader, intended to confuse or dilute information collected by eavesdroppers. By negotiating a shared secret, these commands could be filtered, or "winnowed", by tags using a simple MAC. This procedure is detailed in [84].

5.5.3 Detection Units

RFID-enabled environments may be equipped with devices to detect unauthorized reads or anomalous transmissions on tag operating frequencies. Due to the strong forward signal strength, detecting other readers is a simple matter. Incorporating unauthorized query and jamming detection units into smart shelves would help detect and identify denial of service attacks in a retail setting.

5.5.4 Screaming Tags

Detection units may be extended to detect when tags are disabled. An idea proposed by Sarma [90], is to design tags that "scream" when killed. A scream could be a signal burst on a certain frequency. Incorporating scream detection into smart shelves would further aid in detecting denial of service attacks.

5.5.5 Security Agents

Detection units could be incorporated as a standard feature on all RFID readers, perhaps even into cell phones or PDAs. A legitimate reading device could detect, log and filter other readers' query attempts. A reader could act like a "bouncer" - screening "bad" reads but letting "good" reads pass. In essence, readers and tags would be a Wireless Personal Area Network (WPAN) [47]. Readers would act as a gateway between the WPAN and the outside world.

A locally-held device could even emulate nearby tag contents, eliminating the need to query tags at all and extending their effective range. Juels, Rivest and Szydlo propose a similar approach in [53]. Their idea is to use a "blocker tag" to simulate many ordinary RFID tags and effectively block unauthorized readers.

5.5.6 Printed Master Key

To enable end users to access the functionality of tags affixed to items they have purchased, a master key could be printed within a product's packaging, possibly as a barcode or decimal number. A similar mechanism is proposed for banknotes in [52]. After purchasing an item, a consumer could use the master key to toggle a tag from the hash lock mode of Section 5.1 to the randomized mode of Section 5.2. The master key may also function as a key recovery mechanism, allowing users to unlock tags they have lost the keys to. Since the master key must be read optically from the interior of a package, adversaries cannot obtain it without obtaining the package itself. For further security, all functions using the master key could be required to use a physical contact channel, rather than RF.

Chapter 6

Policy Suggestions

The security and privacy proposals discussed in Chapter 5 are most effective when implemented in conjunction with a well-formed policy. Due to the implementation costs, manufacturers will resist including security features on tags unless pressured by customers or legally obligated to do so. Already, a major clothing retailer succumbed to consumer protests [22] and dropped plans to include insecure RFID tags in their clothing [6].

A proper balance must be struck between protecting consumer privacy and imposing prohibitive costs on manufacturers. Some groups have already called for an outright ban on RFID technology [23]. This reaction is overkill and would deny the cost savings of RFID that would be passed on to consumers. A more judicious policy would allow manufacturers flexibility in tag designs, while guaranteeing basic privacy rights to consumers. Fortunately, there are many related precedents to guide RFID policy decisions.

A reasonable definition of privacy is necessary to consider various policy choices. However, privacy is a notoriously difficult concept to define. The United Nations codifies that "no one shall be subjected to arbitrary interference with his privacy" as a basic human right [103]. It has even been suggested that "all human rights are aspects of privacy" [105].

In 1890, Supreme Court Justice Louis Brandeis famously articulated privacy as the "right to be left alone" [106]. Ruth Gavison of the Yale Law Journal defined three core aspects to privacy: secrecy, anonymity and solitude [42]. The Electronic Privacy Information Center [34] divides privacy into four separate but related categories [18]:

1. **Information Privacy:** Involves rights regarding the handling of personal information such as tax, medical or purchase records. Also known as "data privacy".

2. **Bodily Privacy:** Concerns the right not to be subjected to invasive bodily procedures such as cavity searches and blood, urine or genetic tests.

3. **Communication Privacy:** The right to communicate with others in secrecy.

4. **Territorial Privacy:** Rights limiting intrusion into domestic, workplace or public environments, including searches, identification checks and video surveillance.

The threats discussed in the context of RFID violate some of these rights. An attacker capable of querying tags or eavesdropping on communications clearly violates one's information privacy and Gavison's notion of secrecy. Location privacy issues fall under the category of territorial privacy and are akin to video surveillance.

RFID privacy issues have several distinct properties. Consumers may not even be aware they are carrying RFID tags. For example, a major tire manufacturer recently announced plans to embed RFID tags in their products [80]. Unless notified, most people purchasing tires will never know or even suspect that their car could be tracked by transponders in its tires. Security mechanisms and policy should protect "oblivious" users from exploitation.

Consumers should have some viable option to RFID-tagged products or be able to opt-out without punishment. Otherwise, consumers concerned about privacy might choose a competitor's non-RFID tagged product. A viable option could either be the right to physically remove or destroy tags, or at least the right to a viable alternative. A RFID tollbooth or subway turnstile system must maintain a cash-only alternative which is not significantly impaired. Stores must accept items returned with disabled RFID tags. Consumers should also know what is stored on RFID tags they own and, if possible, when someone is attempting to read it. Furthermore, customers should have access to database records associated with their tags.

Considering these issues, Simson Garfinkel has written an "RFID Bill of Rights" [41] based on the US Department of Health and Education's Code of Fair Information Practices [104]. Garfinkel's Bill of Rights reads as follows:

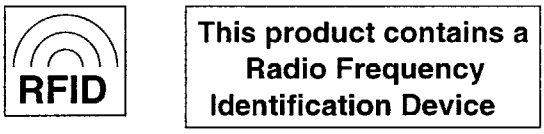

Figure 6-1: Examples of labels which could appear on RFID-labeled products.

" Users of RFID systems and purchasers of products containing RFID tags have:

1. The right to know if a product contains an RFID tag.
2. The right to have embedded RFID tags removed, deactivated or destroyed when a product is purchased.
3. The right to first class RFID alternatives: consumers should not lose other rights (e.g. the right to return a product or to travel on a particular road) if they decide to opt-out of RFID or exercise a RFID tag's "kill" feature.
4. The right to know what information is stored inside their RFID tags and what information is associated with those tags in associated databases. If this information is incorrect, there must be a way to correct or amend it.
5. The right to know when, where and why an RFID tag is being read."

The first right may be satisfied using an industry standard logo on all consumer goods containing a RFID device. Similar logos exist for organic, genetically-modified or irradiated products. This logo would not be a "warning", but simply full disclosure of a product's contents. This allows consumers to make an informed decision and decide whether to disable the tag or purchase an buy an alternative. These choices would be guaranteed by the second and third rights.

Practically speaking, these measures may be easily adopted for low-cost RFID tags. The baseline tag design from Chapter 4 was assumed to have a kill feature. A logo, such as the one depicted in Figure 6-1, could easily be incorporated into most packaging.

Part of Garfinkel's fourth right could be implemented using a physical contact channel to imprint tags [97]. Recall that imprinting was a slow, physical process for setting tag

ownership. Upon purchasing a tagged product, the buyer could use the imprint function to set herself as the tag's new owner. Ownership could have different meanings. In the context of a personal product, ownership might imply having complete control over all tag functionality. For a rented product, "ownership" might only be the right to read a tag. Access to database records associated with tags could be guaranteed in legislation similar to the Fair Credit Reporting Act [20] or the Freedom of Information Act [21].

Garfinkel's fifth right may be broken into two categories. One is controlling what has been read from a tag. The second category is detecting when a tag is read. Access control mechanisms, such as those in Section 5.1, may prevent others from obtaining tag contents. Section 5.2 provides a mechanism that, despite allowing tags to be read at any time, ensures that reads are unpredictable to prevent tracking.

However, instantly notifying a consumer when a read operation has occurred is more difficult. Suggesting of including LEDs or audio speakers as external indicators are impractical and costly. Tags cannot log all read attempts, since they have limited memories or may be read-only.

A solution discussed in Sections 5.5.3 and 5.5.5 is to equip tag readers with detection units so that they may act as security agents. Juels, Rivest and Szydlo propose a similar idea using "blocker tags" [53]. A consumer's RFID-enabled cell phone or PDA could monitor, log and filter all read attempts to his or her tags. Unfortunately, this solution is limited only to those people who own their own portable reading devices. A partial solution might be to include a read counter within tags. This would at least notify consumers of the number of read attempts which occurred in a given period.

Implementation of these standards could be legislated, although that seems to be difficult to enforce. Any legal requirements would require a narrow definition of RFID devices. A manufacturer could circumvent these restrictions with a creative legal team, or by including some other functionality into its RFID tags and marketing them under some other name. Regardless, cheap non-compliant tags might easily be imported. Ideally, voluntary conformance could be enforced through licensing of logos, intellectual property and the pressure of large retailers.

Chapter 7

Open Areas

7.1 Identify Friend or Foe and List Intersection

The hash lock based solutions to access control presented in Sections 5.1 and 5.2 exemplify a trade-off between performance and privacy. Simple hash locked tags offer data privacy, but offer no location privacy. While locked, a tag always responds with the same metaID. Each tag response is identical and allows tags to be tracked. Randomized hash locks address this issue by relying on brute force calculation on the reader side.

This leads to a philosophical question of whether there exists efficient means to identify oneself to friends while revealing no information to enemies. We refer to this as the "Identify Friend or Foe" (IFF) problem. Interestingly, the original RFID application of aircraft transponders [88] ran into this very problem.

Suppose you are a pilot and detect another plane on the radar. It is important to know whether this is a friendly or hostile plane. Both pilots could wait for visual confirmation, but that is a risky gamble and would be susceptible to trickery. Alternatively, you could just radio the other pilot outright with your callsign. If the other pilot were an enemy, this would give them a major advantage. By your callsign, the enemy might know your type of plane or flying style. They could also respond with another friendly callsign to spoof an allied plane. To address this system, the military has developed IFF systems for use on ships, airplanes and missiles. For obvious reasons, much of this work remains classified.

This problem has been studied in other contexts. A broader way of defining the IFF

problem is as the List Intersection Problem. In the List Intersection Problem, two parties, Alice and Bob, each have a list of values A and B. Alice and Bob wish to discover which items are in both lists, that is, $C = A \cap B$. An eavesdropper Eve should learn nothing of A, B or C. A malicious attacker Mallory, pretending that she is Bob, will only learn some set $A \cap M$. Assuming that $A \subseteq S$ and $B \subseteq S$, where $|A|$ and $|B|$ are negligible with respect to $|S|$, Mallory learns negligible information about Alice's list.

The RFID identification problem is a special case of the List Intersection Problem called the One-to-Many Intersection Problem. In this case, Alice has a single value a while Bob has many values B. Bob learns the value of a and Alice learns that $a \in B$. Eve learns nothing.

Naor and Pinkus offer an efficient solution to the List Intersection Problem in [69]. Their solution reduces the general List Intersection Problem to many invokations of the One-to-Many Intersection Problem, which is in turn solved using Oblivious Polynomial Evaluation.

In the Oblivious Polynomial Evaluation problem, Alice knows some polynomial P. Bob wishes to compute $P(x)$. However, Bob should learn nothing about P other than $P(x)$ and Alice should learn nothing of x. Oblivious Polynomial Evaluation relies on Oblivious Transfer [76] as a building block.

Oblivious transfer protocols may rely on base assumptions involving Diffie-Hellman [70], channel noise [26] or quantum properties [25]. These solutions are too inefficient or are inapplicable to RFID devices. However, leveraging channel noise for privacy amplification [8] may be an intriguing possibility for RFID.

The List Intersection Problem and its relation to RFID leads to two open questions. First, are there oblivious transfer protocols appropriate for low-cost RFID devices? Second, are there other efficient solutions to the List Intersection problem relying on other fundamental building blocks? Are their other solutions to the Identify Friend or Foe problem which do not rely on the List Intersection problem?

The dependencies of the problems and solutions discussed in this section are depicted as a tree in Figure 7-1. The IFF problem is at the root of the tree. It may be addressed using randomized hash locks or reduced to a List Intersection Problem, which are shown

as children of the IFF node. Subsequently dependencies between problems are also shown.

7.2 Protocols

New RFID protocols resistant to eavesdropping, fault induction and power analysis need to be developed. The anti-collision algorithms presented in Section 5.4 offer protection against long range eavesdropping, but are still vulnerable to nearby eavesdroppers and fault induction. The Blinded Tree-Walking protocol also requires that a population of tags share a common prefix unknown to eavesdroppers, which is not always a valid assumption. Randomized Tree-Walking eliminates this assumption, but at the cost of performance. In general, readers and tags must be designed to gracefully recover from interruption or fault induction without compromising security.

7.3 Hardware

RFID security and privacy will greatly benefit from the continued development of hardware-efficient cryptographic hash functions, symmetric encryption, message authentication codes and random number generators. Section 5.3 specified two promising avenues for low-cost hardware hash function design: cellular automata and non-linear feedback shift registers.

General advances in circuit fabrication and RFID manufacturing will lower costs and allow more resources to be allocated for security features. Continued research into efficient symmetric encryption algorithms, such as TEA [109, 110], may yield algorithms appropriate for low-cost RFID devices. One open question from Section 5.2 is whether pseudo-random function ensembles can be implemented with significantly less complexity than symmetric encryption. Designing efficient implementations of perfect one-way functions [16] may be a relevant avenue of research as well.

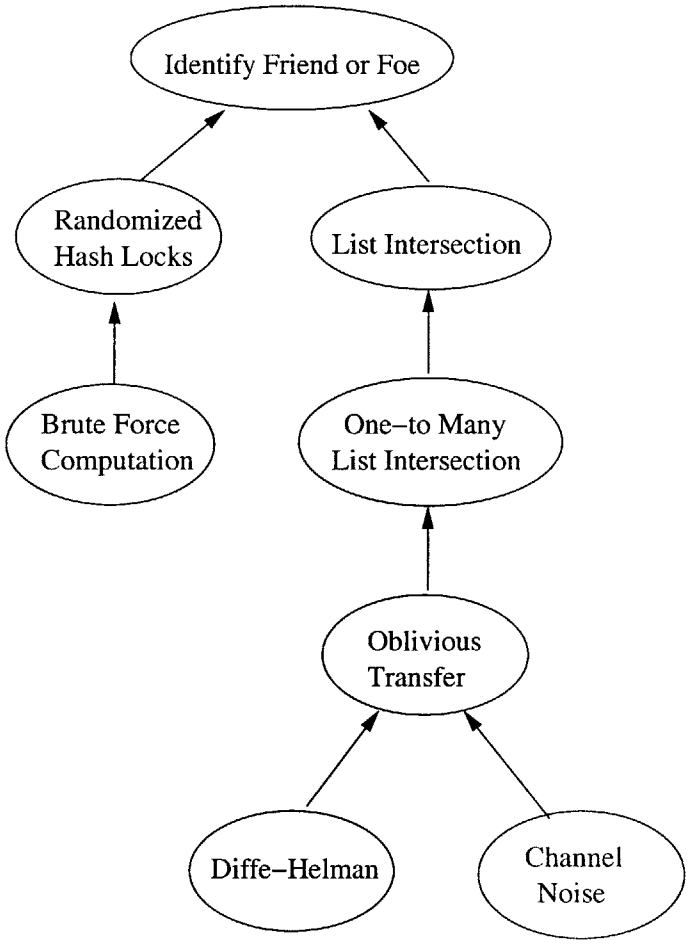

Figure 7-1: A dependency tree for solutions to the Identify Friend or Foe Problem.

Appendix A

Glossary

active tags An RFID tag with an on-board power source, such as a battery, and an active transmitter, 17.

asymmetric channel strength The differential between signal strength between the forward and backward channels; a security issue in passively powered tags, page 34.

auto-ID Automatic Identification, page 8.

backward channel The communications channel from tag to reader, page 19.

cellular automata (CA) A set of finite state machines whose state transitions depend on nearby neighbors, page 46.

collision Interference resulting from simultaneous transmissions on the same frequency, page 24.

collision-resistant hash function A hash function which is 2nd-preimage and collision resistant, 44.

decoy attack Replacing a valid RFID tag with a counterfeit, page 30.

EPC Electronic Product Code, page 14.

far-field The range outside $\frac{1}{2\pi}$ a signal's wavelength, page 21.

feedback shift register A shift register coupled with a feedback function, page 47.

forward channel The communications channel from reader to tag, page 19.

gate density Gates per mm^2 of silicon, dependent on line width technology, page 33 and [7].

hash function An efficiently computable function which maps an arbitrary length input to a fixed length output, page 43 and [64].

hash lock Access control mechanism specified in Sections 5.1 and 5.2, page 37.

IC integrated circuit, page 14.

IFF Identity Friend or Foe, page 61.

imprinting The process of setting ownership of a new device, page 35 and [97].

inductive coupling Process of obtaining power from a current induced by a magnetic field, page 21.

ISM Industrial-Scientific-Medical bands, used by low-power, short-range RF systems, page 25 and [50].

level code An encoding scheme in which binary values are represented by specific voltage levels, page 22.

LFSR Linear Feedback Shift Register, page 49.

location privacy The ability to prevent other parties from learning one's current and past locations, page 28.

low-cost tags RFID tags priced in the US$0.50-US$0.10 (5-10¢) range, page 14.

metaID The hashed key value which acts as a temporary ID for a tag, defined in Section 5.1, page 38.

near-field The range within $\frac{1}{2\pi}$ a signal's wavelength, page 21.

NLFSR Non-Linear Feedback Shift Register, page 49.

nonce A random string used to pad messages, used in Section 5.2, page 40.

one-way hash function A hash function offering preimage and 2nd-preimage resistance, 43.

passive tags An RFID tag which receives power from a reader, necessarily with a passive transmitter, page 17.

RFID Radio-Frequency Identification, page 10.

semi-passive tags An RFID tag with an on-board power source, but with a passive transmitter, page 17.

side band The frequency spread of a signal resulting from modulation, page 21.

singulation Addressing and isolating a single tag from among a population of many tags; typically by means of an anti-collision algorithm, page 25.

smart shelf A shelf outfitted with a fixed RFID reader used in automated inventory control systems, page 19.

transition code An encoding scheme in which binary values are represented by transitions between voltage levels, page 22.

UPC Universal Product Code, page 9 and [102].

Appendix B

Cast of Characters

Alice A legitimate player in cryptographic protocols.

Bob A legitimate player in cryptographic protocols.

Denise An attacker limited to conducting denial of service attacks.

Eve An eavesdropper only capable of passive monitoring.

Mallory A malicious adversary who may impersonate legitimate players.

Phyllis An attacker who may attack the physical hardware of a system.

Tracy An adversary who cannot eavesdrop, but can tell when messages are sent.

Bibliography

[1] Martin Abadi, Michael Burrows, C. Kaufman, and Butler W. Lampson. Authentication and Delegation with Smart-cards. In *Theoretical Aspects of Computer Software*, pages 326–345, 1991.

[2] Alma Technologies. SHA-1 Cores. http://www.alma-tech.com.

[3] Ross Anderson and Markus Kuhn. Low Cost Attacks on Tamper Resistant Devices. In *IWSP: International Workshop on Security Protocols, LNCS*, 1997.

[4] Auto-ID Center. Draft Protocol Specification for a Class 0 Radio Frequency Identification Tag, February 2003.

[5] S. Bakhtiari, R. Safavi-Naini, and J. Pieprzyk. Cryptographic Hash Functions: A Survey. Technical Report 95-09, Department of Computer Science, Universit of Wollongong, July 1995.

[6] Elisa Batista. A 'Step Back' for Wireless ID Tech? *Wired Magazine*, April 2003.

[7] Lucas Bauer and Otto Manck. Perspectives of Modern ASIC Design. In *Symposium on Opto- and Microelectronic Devices and Circuits*, March 2002.

[8] Charles H. Bennett, Gilles Brassard, Claude Crépeau, and Ueli Maurer. Generalized Privacy Amplification. *IEEE Transaction on Information Theory*, 41(6):1915–1923, 1995.

[9] Alastair Beresford and Frank Stajano. Location Privacy in Pervasive Computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[10] Eli Biham and Adi Shamir. Differential Cryptanalysis of FEAL and N-Hash. In D.W. Davies, editor, *Advances in Cryptology - EUROCRYPT*, LNCS, pages 1–16. Springer-Verlag, 1991.

[11] Benny Bing. *Broadband Wireless Access*. Kluwer Academic Publishers, 2002.

[12] John Black, Phillip Rogaway, and Thomas Shrimpton. Black-Box Analysis of the Block-Cipher-Based Hash Function Constructions from PGV. In *Advances in Cryptology - CRYPTO*, LNCS. Springer-Verlag, 2002.

[13] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the Importance of Checking Cryptographic Protocols for Faults. In *EUROCRYPT'97*, volume 1233, pages 37–51. Lecture Notes in Computer Science, Advances in Cryptology, 1997.

[14] Antoon Bosselaers, René Govaerts, and Joos Vandewalle. Fast Hashing on the Pentium. In *Advances in Cryptology - CRYPTO*, volume 1109 of *LNCS*, pages 298–313. Springer-Verlag, 1996.

[15] Paul Camion and Jacques Patarin. The Knapsck Hash Function Proposed at Crypto '89 Can be Broken. In *Advances in Cryptology - EUROCRYPT*, pages 39–53, 1991.

[16] Ran Canetti, Daniele Micciancio, and Omer Reingold. Perfectly One-Way Probabilistic Hash Functions. In *30th Annual ACM Symposium on Theory of Computing*, pages 131–140, 1998.

[17] CAST Inc. AES and SHA-1 Cryptoprocessor Cores. http://www.cast-inc.com.

[18] Electronic Privacy Information Center. *Privacy and Human Rights: An International Survey of Privacy Laws and Developments*. EPIC.org, 2002.

[19] Suresh Chari, Charanjit Jutla, Josyula R. Rao, and Pankaj Rohatgi. A Cautionary Note Regarding Evaluation of AES Candidates on Smart-Cards. In *Second Advanced Encryption Standard (AES) Candidate Conference*, Rome, Italy, 1999.

[20] United States Code. Fair Credit Reporting Act. US Code Title 15 Chapter 41 Section 1681.

[21] United States Code. Freedom of Information Act. US Code Title 5 Chapter 5 Section 552.

[22] Consumers Against Supermarket Privacy Invasion and Numbering. Boycott Benetton. http://www.boycottbenetton.org/.

[23] Consumers Against Supermarket Privacy Invasion and Numbering. CASPIAN Website. http://www.nocards.org/.

[24] Jim Crane. Benetton Clothing to Carry Tiny Tracking Transmitters. Associated Press, March 2003.

[25] Claude Crépeau. Quantum Oblivious Transfer. *Journal of Modern Optics*, 41(12):2455–2466, 1994.

[26] Claude Crépeau and Joe Kilian. Achieving Oblivious Transfer Using Weakened Security Assumptions. In *Foundations of Computer Science (FOCS*, pages 42–52, 1988.

[27] Joan Daemen, René Govaerts, and Joos Vandewalle. Hash Functions Based on Block Ciphers: A Synthetic Approach. In *Advances in Cryptology - ASIACRYPT*, LNCS. Springer-Verlag, 1991.

[28] Joan Daemen, René Govaerts, and Joos Vandewalle. A Hardware Design Model for Cryptographic Algorithms. In *European Symposium on Research in Computer Security*, pages 417–434, 1992.

[29] Ivan Damgard. A Design Principle for Hash Functions. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO*, volume 435 of *LNCS*, pages 416–427. Springer-Verlag, August 1989.

[30] Hans Dobbertin. Alf Swindles Ann. *CryptoBytes (RSA Laboratories)*, 1(3):5, Autum 1995.

[31] Hans Dobbertin, Antoon Bosselaers, and Bart Preneel. RIPEMD-160, A Strengthened Version of RIPEMD. In D. Gollmann, editor, *Fast Software Encryption*, volume 1039 of *LNCS*, pages 71–82. Springer-Verlag, 1996.

[32] E-Z Pass. Website. http://www.ezpass.com.

[33] EAN International and the Uniform Code Council. http://www.ean-int.org.

[34] Electronic Privacy Information Center. EPIC Website. http://www.epic.org.

[35] Omega Electronics. RFID Swatch Watch. http://www.omega-electronics.ch/rfid/swatch.shtml.

[36] Taher ElGamal. A Public Key Cryptosystem and Signature Scheme Based on Discrete Logarithms. *IEEE Trans. Information Theory*, 31(4):469–472, July 1985.

[37] Federal Information Processing Standards (FIPS). Data Encryption Standard. Technical Report 46-2, National Institute of Standards and Technology (NIST), January 1988. supersedes FIPS PUB 46-1, 1977.

[38] Federal Information Processing Standards (FIPS). Secure Hash Standard (SHA-1). Technical Report 180-1, National Institute of Standards and Technology (NIST), April 1995. supersedes FIPS PUB 180, 1993.

[39] Federal Information Processing Standards (FIPS). Advanced Encryption Standard. Technical Report 197, National Institute of Standards and Technology (NIST), November 2001.

[40] R. Fletcher, O. Omojola, E. Boyden, and N. Gershenfeld. Reconfigurable Agile Tag Reader Technologies for Combined EAS and RFID Capability. In *Workshop on Automatic Identification Advanced Technologies*, October 1999.

[41] Simson L. Garfinkel. Adopting Fair Information Practices in Low-Cost RFID Systems. In *Ubiquitious Computing*, September 2002.

[42] Ruth Gavison. Privacy and the Limits of Law. *Yale Law Review*, pages 421–428, 1980.

[43] Howard Gobioff, Sean Smith, J. Doug Tygar, and Bennet Yee. Smart Cards in Hostile Environments. In *2nd USENIX Workshop on Elec. Commerce*, 1996.

[44] Solomon W. Golomb. *Shift Register Sequences*. Holden-Day, 1967.

[45] Samil Harari. Non-Linear, Non-Commutattive Functions for Data Integrity. In *Advances in Cryptology - EUROCRYPT*, pages 25–32, 1984.

[46] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A Ring-Based Public Key Cryptosystem. *Lecture Notes in Computer Science*, 1423:267–, 1998.

[47] IEEE. 802.15: Wireless Personal Area Network (WPAN) Working Group. http://grouper.ieee.org/groups/802/15.

[48] International Standards Organization. ISO/IEC 9797: Data Integrity Mechanism Using A Cryptographic Check Function Employiung a Block Cipher Algorithm. http://www.iso.org, 1989.

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

[50] International Telecommunications Union. Radio Regulations, 1998. Volume 1.

[51] Markus Jakobsson and Susanne Wetzel. Security Weaknesses in Bluetooth. *Lecture Notes in Computer Science*, 2020:176+, 2001.

[52] Ari Juels and Ravikanth Pappu. Squealing Euros: Privacy Protection in RFID-Enabled Banknotes. In *Financial Cryptography*, 2002.

[53] Ari Juels, Ronald L. Rivest, and Michael Szydlo. The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy. http://theory.lcs.mit.edu/~ rivest/, May 2003. Submitted for publication.

[54] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next Century Challenges: Mobile Networking for "Smart Dust". In *MOBICOM*, pages 271–278, 1999.

[55] Burton Kaliski. The MD2 Message Digest Algorithm. Technical Report RFC 1319, RSA Laboratories, April 1992.

[56] Burton S. Kaliski Jr and Matt J. B. Robshaw. Comments on Some New Attacks on Cryptographic Devices. RSA Laboratories' Bulletin No. 5, July 1997. http://www.rsasecurity.com/rsalabs/bulletins/.

[57] Neal Koblitz. Elliptic Curve Cryptosystems. *Mathematics of Computation*, 48(177):203–209, 1987.

[58] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. *Lecture Notes in Computer Science*, 1666:388–397, 1999.

[59] Paul C. Kocher. Cryptanalysis of Diffie-Hellman, RSA, DSS, and other Systems Using Timing Attacks. Technical report, Cryptography Research, Inc., 1995.

[60] Alfred R Koelle, Steven W. Depp, Jermy A. Landt, and Ronald E. Bobbett. Short-Range Passive Telemetry by Modulated Backscatter of Incident CW RF Carrier Beams. *Biotelemetry*, 3:337–340, 1976.

[61] Matthias Krause and Stefan Lucks. On the Minimal Hardware Complexity of Pseudorandom Function Generators. In *Theoretical Aspects of Computer Science*, volume 2010, pages 419–435. Lecture Notes in Computer Science, 2001.

[62] Xuejia Lai, Rainer Rueppel, and Jack Wooliven. A Fast Cryptographic Checksum Algorithm Based on Stream Ciphers. In *Advances in Cryptology - AUSCRYPT 92*, volume 718 of *LNCS*, pages 339–348. Springer-Verlag, 1992.

[63] Michael Luby and Charles Rackoff. How to Construct Pseudorandom Permutations from Pseudorandom Functions. *SIAM Journal on Computing*, 17(2):373–386, April 1988.

[64] Alfred J. Menezes, Paul C. van Oorshot, and Scott A. Vanstone. *Handbook of Applied Cryptography*, chapter 1.9. CRC Press, 1996.

[65] Ralph Merkle and Marty Hellman. Hiding Information and Signatures in Trapdoor Knapsacks. *IEEE Trans. Information Theory*, 24:525–530, September 1978.

[66] Robert M. Metcalfe and David R. Boggs. Ethernet: Distributed Packet Switching for Local Computer Networks. *Communications of the ACM*, 19(5):395–404, July 1976.

[67] MIT. Auto-ID Center. http://www.autoidcenter.org.

[68] Mobile Cloak. Website. http://www.mobilecloak.com.

[69] Moni Naor and Benny Pinkas. Oblivious Transfer and Polynomial Evaluation. In *Symposium on Theory of Computer Science (STOC)*, pages 245–254, May 1999.

[70] Moni Naor and Benny Pinkas. Efficient Oblivious Transfer Protocols. In *Symposium on Discrete Algorithms (SODA)*, January 2001.

[71] National Security Agency. TEMPEST Fundamentals. Technical report, National Security Agency, February 1982. Released under FOIA: http://cryptome.org/nacsim-5000.htm.

[72] NTRU. GenuID. http://www.ntru.com/products/genuid.htm.

[73] Greg Papadopoulos. Finishing the revolution. MIT Dertouzos Lecture Series, February 2003.

[74] Bart Preneel. *Analysis and Design of Cryptographic Hash Functions*. PhD thesis, Katholieke University Leuven, January 1993.

[75] Bart Preneel, René Govaerts, and Joos Vandewalle. Hash Functions Based on Block Ciphers: A Synthetic Approach. In *Advances in Cryptology - CRYPTO*, LNCS, pages 368–378. Springer-Verlag, 1994.

[76] Michael Rabin. How to Exchange Secrets by Oblivious Transfer. Memo TR-81, Aiken Computation Laboratory, Harvard University, 1981.

[77] re-code.com. Corporations Win Again! http://www.re-code.com, April 2003.

[78] RFID Journal. Gillette to Purchase 500 Million EPC Tags. http://www.rfidjournal.com, November 2002.

[79] RFID Journal. Learning from Prada. http://www.rfidjournal.com/article/articleview/272, June 2002.

[80] RFID Journal. Michelin Embeds RFID Tags in Tires. http://www.rfidjournal.com, January 2003.

[81] Ronald L. Rivest. The MD4 Message Digest Algorithm. In *Advanced in Cryptology - CRYPTO*, pages 303–311. Springer-Verlag, 1990.

[82] Ronald L. Rivest. The MD4 Message Digest Algorithm. Technical Report RFC 1320, MIT Lab for Computer Science and RSA Laboratories, April 1992.

[83] Ronald L. Rivest. The MD5 Message Digest Algorithm. Technical Report RFC 1321, MIT Lab for Computer Science and RSA Laboratories, April 1992.

[84] Ronald L. Rivest. Chaffing and Winnowing: Confidentiality without Encryption. *CryptoBytes (RSA Laboratories)*, 4(1):12–17, Summer 1998.

[85] Ronald L. Rivest. Personal correspondance. May 2003.

[86] Ronald L. Rivest, Adi Shamir, and Len Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[87] N. Rogier and Pascal Chauvaud. The Compression Function of MD2 is Not Collision Free. In *Selected Areas in Cryptography*, May 1995.

[88] Royal Air Force. History: 1940. http://www.raf.mod.uk/history/line1940.html.

[89] Sanjay E. Sarma. Towards the 5¢ Tag. Technical Report MIT-AUTOID-WH-006, MIT Auto-ID Center, February 2001.

[90] Sanjay E. Sarma. Personal correspondance. 2002.

[91] Sanjay E. Sarma, Stephen A. Weis, and Daniel W. Engels. RFID Systems and Security and Privacy Implications. In *Workshop on Cryptographic Hardware and Embedded Systems*, pages 454–470. Lecture Notes in Computer Science, 2002.

[92] Sanjay E. Sarma, Stephen A. Weis, and Daniel W. Engels. Radio Frequency Identification: Risks and Challenges. *CryptoBytes (RSA Laboratories)*, 6(1), Winter/Spring 2003.

[93] Tom Ahlkvist Scharfeld. An Analysis of the Fundamental Constraints on Low Cost Passive Radio-Frequency Indentification System Design. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA 02139, August 2001.

[94] Winn Schwartau. *Information Warfare*, chapter HERF Guns and EMP/T Bombs, pages 118–129. Thunder's Mouth, 1994.

[95] Ernst Selmer. *Linear Recurrence over Finite Field.* University of Bergen, 1966.

[96] Claude Shannon. Communication Theory of Secrecy Systems. *Bell Systems Technical Journal*, 28(4):656–715, 1949.

[97] Frank Stajano and Ross Anderson. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In *7th International Workshop on Security Protocols*, volume 1796, pages 172–194. Lecture Notes in Computer Science, 1999.

[98] Stamps.com. Homepage. http://www.stamps.com.

[99] Roger Stewart. Personal correspondance. CTO Alien Technology Corporation, December 2002.

[100] TAMPER Lab. University of Cambridge Tamper and Monitoring Protection Engineering Research Lab. http://www.cl.cam.ac.uk/Research/Security/tamper.

[101] J. Doug Tygar and Bennet Yee. Cryptography: It's Not Just For Electronic Mail Anymore. Technical Report CS-93-107, Carnegie Mellon University, 1993.

[102] Uniform Code Council. Homepage. http://www.uc-council.org.

[103] United Nations. Universal Declaration of Human Rights. *General Assembly Resolution*, 217 A(III), December 1948.

[104] US Dept. of Health Education and Welfare. Secretary's Advisory Committee on Automated Personal Data Systems, Records Computers and the Rights of Citizens, 1973.

[105] Fernando Volio. *The International Bill of Rights: The Covenant on Civil and Political Rights*, chapter Legal Personality, Privacy and the Family. Columbia University Press, 1981.

[106] Samuel Warren and Louis Brandeis. The Right to Privacy. *Harvard Law Review*, pages 193–220, 1890.

[107] Steve H. Weigart. Physical Security Devices for Computer Subsystems: A Survey of Attacks and Defences. In *Workshop on Cryptographic Hardware and Embedded Systems*, volume 1965, pages 302–317. Lecture Notes in Computer Science, 2000.

[108] Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest, and Daniel W. Engels. Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In *Security in Pervasive Computing*, 2003.

[109] David J. Wheeler and Robert M. Needham. TEA, a Tiny Encryption Algorithm. Technical report, Computer Laboratory, University of Cambridge, 1995.

[110] David J. Wheeler and Robert M. Needham. TEA Extensions. Technical report, Computer Laboratory, University of Cambridge, 1997.

[111] Stephen Wolfram. Cryptography with Cellular Autonoma. In *Advances in Cryptology - CRYPTO*, volume 218 of *LNCS*, pages 429–432. Springer-Verlag, 1985.

[112] Stephen Wolfram. Random Sequence Generation By Cellular Automata. *Advances in Applied Mathematics*, 7:123–169, June 1986.

[113] Stephen Wolfram. *A New Kind of Science*. Wolfram Media, 2002.

[114] Gilles Zémor. Hash Functions and Graphs with Large Girths. In *Advances in Cryptology - EUROCRYPT*, pages 506–511, 1991.

[115] Yuliang Zheng, Josef Pieprzyk, and Jennifer Seberry. HAVAL – A One-way Hashing Algorithm with Variable Length of Output. In *Advances in Cryptology - AUSCRYPT*, LNCS. Springer-Verlag, 1990.