Survey of Cryptographic Implementations and Vulnerabilities on Passive RFID Devices

Pedro Miguel Sosa
UCSB, Santa Barbara, CA - USA

June 8, 2016

Abstract

Passive RFID devices (tags) have become essentially ubiquitous in today's world. We find them in our credit cards, key cards, and many other consumer products. Many companies in the past have claimed that their RFID-enabled products are secure, however this industry has consistently opted to maintain a *security by obscurity* approach, using closed proprietary cryptographic solutions. This approach has failed multiple times, as researchers and hackers (both white-hats and black-hats alike) have reverse-engineered and exploited their products. This survey paper aims to give a solid groundwork for researchers wishing to understand the goals and challenges in the area of cryptographically-enabled passive RFID devices.

In this paper we will provide the basic models of passive RFID device usages, and a series of common attack models. Furthermore we will study two famous cases of RFID reverse-engineering and their implications. Finally, we will survey possible techniques for mitigating typical vulnerabilities.

1 Introduction

1.1 What is RFID?

Radio-Frequency Identification (RFID) is a wireless technology intended for the automated identification of objects. While the RFID technology spectrum is actually quite broad, these devices can be roughly divided into tags and interrogators. Tags consist of small passive microchips connected to an antenna. These devices don't have a power source of their own, instead they are designed to work with energy collected by a nearby RFID reader (interrogator). The RFID interrogator, on the other hand, does have an internal

power source and is usually connected to some external backend (server or database). Essentially, tags receive a query from the interrogator, run some computation or check memory, and return a response.

1.2 RFID in the Wild: How are they used?

The simplicity and versatility of this device has made it extremely popular throughout this last decade. With a market veering towards $10 billion in 2015 [1], we find that RFID has become ubiquitous in today's world. While RFID is mostly intended for identification, its has grown to support more complex usages. Some prime examples of modern day usage are:

- **Contact-less credit cards** (e.g. SpeedPass, Mastercard, American Express) allow to pay without actually swiping cards.
- **Automobile Keys** Some automobiles now come with RFID enabled keys as a theft deterrent.
- **Public Transport Payment Fare cards** (e.g. Oyster cards) To keep a digital wallet to manage public transportation payments.
- **Proximity building access cards** Used instead of physical keys.
- **Supply chain identification** Helps tracking products in manufacturing plants.

1.3 Classification of RFID Devices

RFID devices are usually classified in terms of their capabilities which are roughly limited by their passive or active nature. Devices with onboard power sources will generally achieve longer range and support more gates.

EPCglobal is an organization that seeks to define standards within technology. Their classification system for RFID is now widespread and considered standardized. This classification is based on usage and implementation, and consists of 4 RFID classes.

Class-1: Identity Card	Passive tag Electronic Product Code Tag Identifier *Kill* function - permanently disables tag Opt. password-protected access control Opt. user memory
Class-2: Higher-Functionality Tags	Class-1 Features + Passive tag Extended TID Extended User Memory Authentication controls memory
Class-3: Semi-Passive Tags	Class-2 Features + Semi-Passive tag. Integral power source Integrated sensing circuitry
Class-4: Semi-Passive Tags	Class-3 Features + Tag-to-Tag communication Active communications Ad-hoc networking capabilities

Table 1: EPCglobal's classification of RFID devices

This classes don't specify actual frequency standards (and read range), as these regulations are left to the governmental agencies of each country. In general one can differentiate the RFID devices ranges as Low, High, and UltraHigh Frequency. [1]

Frequency	Regulation	Range	Data Speed
LF: 120-150 kHz	Unregulated	10cm	Low
HF: 13.56 MHz	ISM band worldwide	10cm - 1m	Low to Moderate
UHF: 433 MHz	Short Range Devices	1m - 100m	Moderate

Table 2: Frequency classification of RFID devices

Sidenote: NFC As an important side-note, today it is common to see phones or credit cards that come pre-packaged with NFC technologies or capabilities. Near Field Communication (NFC) is a subset of RFID technologies that work on the 13.56MHz HF range and provide features such

[1] Microwave frequencies of 2450-5800MHz and 3.1-10GHz are also defined, yet are rather rare.

as peer-to-peer information transfer, smartcard emulation, and read/write modes. These have become particularly popular due to their versatility when paired up with phones, since they can effectively emulate credit cards to pay (e.g Google Pay), or share information amongst phones. However, this type of device does not fall in the focus of this paper since the cryptographic complexity will depend on the phone's hardware and software solutions.

1.4 Scope

For the following paper will focus on passive RFID tags, roughly equivalent to EPC Class-1 and Class-2. These tags are the most popularly used in security systems (e.g. credit cards, key cards, and virtual wallets). These are also the devices that present the most restrictions in terms of power consumption and circuit complexity.

Unique identifier	Length of 96-256 bits.
Frequencies	HF or UHF.
Read Range	HF: 3m-5m and UHF: 200-500mm.
Number of Gates	7,000 - 10,000.
Read/Sec	200-1500 (user performance requirement).
Power consumption	10s of micro-watts.

Table 3: Characteristics of passive tags (EPC Class-1 and Class-2) [2]

2 Attack Framework

To better understanding the vulnerabilities and possible attacks that could be done against these devices, it is important to understand the overall system in which they work and the types of adversaries that a system like this could encounter.

2.1 System Model

Communication Channels A typical RFID system consists on a reader and a tag. The communication between the reader and tag can be abstracted into 3 channels: power, forward, and backwards. The power channel refers to the microwaves meant to power the tag. The forward channel is used by the reader to send commands to the tag. The backward channel is used by the tag to responds to the reader.

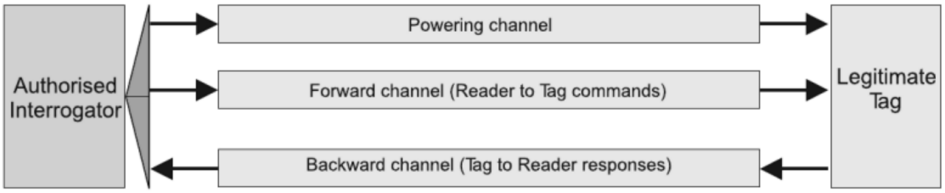

Figure 1: System model (Damith C. Ranasinghe "Confronting Security and Privacy Threats in Modern RFID System", 2006 IEEE)

Singulation To avoid collisions between multiple tags responding to a reader's query, RFID protocol uses singluation. Singulation is a process by which the reader first determines which tags are present and then queries one tag at a time. There are multiple singulation protocols, but the two most common ones are *ALOHA* and *Tree-walking*.

Tree-Walking In tree-walking the reader performs a depth-first search across a tree of ids to identify tags. For example, a reader first queries tags that start with "1" or "0". If it receives a concordant response of "1" or "0" the reader can then can disregard half of the tree and instead proceeds to recursively query the "1X" or "0X" branch respectively. However if the tag receives a discordant collision response, it knows that it has to keep recursing on both halves of the tree. The reader proceeds until it has either talked to all tags, or found the tag it is looking for (depending on the scenario).

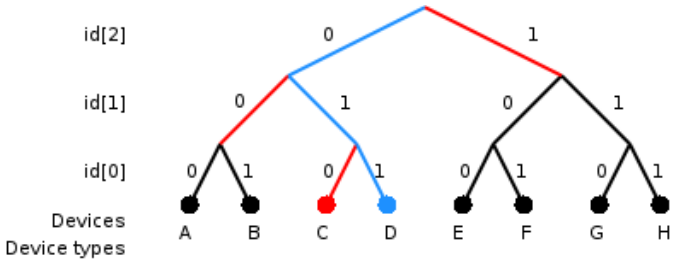

Figure 2: Tree-walking (source: Rob Blanco)

ALOHA The ALOHA protocol is a collide-and-resend protocol, where if a collision is encountered, hosts will wait a random interval and resend the data. There is also a slotted ALOHA version in which time slots are designated.

2.2 Adversary Model

There are multiple types of adversaries that could potentially seek to break an RFID system. It is important to define the motivation and capabilities of the adversary to better understand what types of attack could be done on these devices.

Capabilities One of the main distinctions is based on whether the attacker will interact with the tags or not. A passive attacker can listen in to conversations, but not actively participate in them. An active attacker can choose to participate in a variety of ways, perhaps providing interference (as is the case of a DoS attack), spoofing a legitimate interrogator, or pretending to be a specific tag.

Motivations The most common motivation behind an RFID attack is to clone the ID of a certain object or person. However, other motivations such as modifying tag data or avoiding tag detection are also possible. Figure 3 shows a series of possible RFID interactions that an attacker could attempt.

As a side note, physical attacks on the tag or reader are also possible, however many of these lead to equipment damage so they are less common.

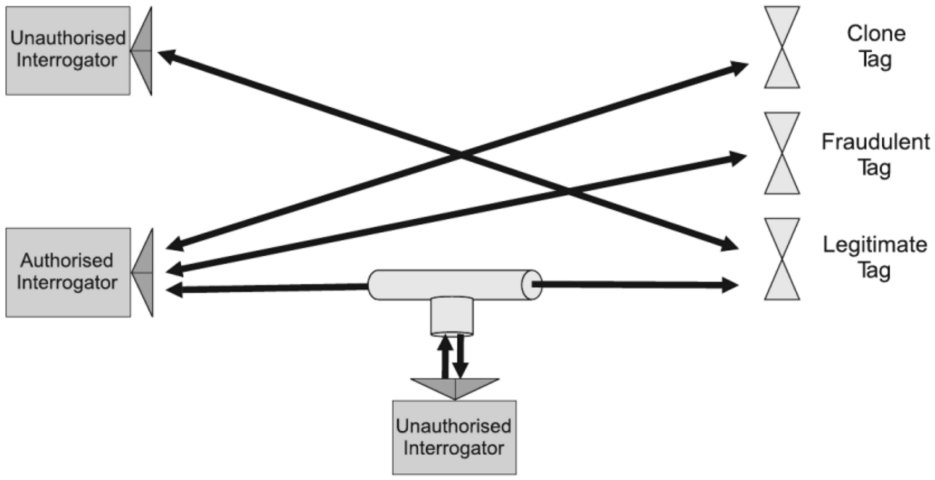

Figure 3: Possible RFID Interactions (Damith C. Ranasinghe "Confronting Security and Privacy Threats in Modern RFID System", 2006 IEEE)

2.3 Attacks

2.3.1 Eavesdropping

While eavesdropping isn't generally considered an attack on its own, it is usually the first step towards a more comprehensive attack. Due to the wireless nature of these devices, any attacker can gather information from them by just being within a given range. Depending on the range limitations an attacker can choose to do an active or passive attack.

Active Eavesdropping Active eavesdropping requires and attacker to use her own reader to query the target tags. Because the attacker must provide power to the these tags, this type of scan has to be done within inches of the target. While this might be perceived as a restriction, its benefit is that an attacker has the ability to choose the queries (chosen-challenge attack), thus allowing her to procure more precise information.

Passive Eavesdropping Passive eavesdropping, on the other hand, allows the attacker to scan a legitimate communication between a valid reader and a given tag. One of the biggest advantages of passive scanning is the increased range of attack, since the range solely depends on the ability to intercept the signals emitted by the devices. The US Department of Homeland Security reported successful eavesdropping of 13.56 Mhz RFID devices from tens of feet [3]. The actual maximum range will change depending on the frequencies used by the tags.

2.3.2 Cloning & Spoofing

The most common RFID attack consists on cloning a legitimate cards to later trick a reader into believing you are in possession of said card. These attacks can be quite harmful in the case of credit cards, car keys, or access cards.

While the methodology of this attack depends on the protocol and cipher in use, in most cases, it is a combination of eavesdropping, secret-key brute-forcing, and card emulation.

There are many different tools today that allow for such attacks. The most popular device used in the field is the Proxmark3. The Proxmark3 is a small RFID reader, writer, and simulator which can be used in conjunction with custom firmware and scripts. Some scripts, such as RFIDiot and ProxBrute allow brute-forcing secret-keys on the fly, checking challenges against rainbow tables, and provide commonly used tag emulation [4].

2.3.3 Power Analysis

Power analysis is a side channel attack that consists on obtaining leaked information from fluctuations in power consumption of RFID Devices.

As with all electronic devices, when RFID circuitry is doing some operation, electricity will be moving around and generating a small magnetic field. The attack relies on reading this magnetic field and using this information to decipher what the logical circuit is doing.

One such attack was described by Oren and Shamir [5]. In this particular attack, the opponent was able to accurately retrieve class-1 tag's kill password. The kill password is part of the functionality that allows a tag to be permanently deactivated. This attack could be potentially useful in a store environment, where an attacker could deactivate a product tag and thus walk away with merchandise without alerting any anti-theft systems.

2.3.4 Relay Attacks

Relay Attacks consists on creating an extended channel between a valid reader and a valid tag. In this case, the extension is accomplished by adding a device that will relay the readers query to another devices which will communicate with the valid tag.

In essence, an attacker would allow an RFID tag (such as a credit card) to speak with a reader (Point of Sales device) regardless of physical distance. This attack has been shown to work particularly well with NFC enabled-phones and *Google Pay* [6].

2.3.5 DoS

Denial of Service attacks on RFID systems are actually surprisingly simple. Most of the time these attacks rely on creating or forcing collisions when readers try to communicate with valid tags.

The simplest attack could consist on using a jamming device which feeds noise into the tag spectrum. This would consistently create collisions, making it impossible for the tag to communicate with the reader, or vice versa.

On some devices, such as key card systems, it sufficient to place an unauthorized tag in close proximity to the reader. Such an attack was demonstrated during CanSecWest [7] where a reader controlling a door mechanism would refuse to grant access if an unauthorized tag was placed in close proximity.

3 Study Cases

We will now study some real-life examples of broken RFID cryptosystems. While there are many different examples that we could use, we will focus on the Mifare Classic (2008) and DST cards (2006) These are historical cases that had the most repercussion and set the standard methodology for breaking RFID devices.

3.1 Mifare Card

Introduction For our first case study, we will focus on one of the most famous cases of RFID reverse engineering: Karsten Nohl's reverse engineering of the Mifare Classic card [8],[9]. The Mifare cards are meant to be digital wallets for public transportation fare money. By 2008, there where more than 2 billion cards in circulation and it was the de-facto card in the subways of many cities such as London, Santiago, and Istanbul. When this paper was published, the card's manufacturer NXP unsuccessfully sued researchers who tried to do further cryptanalysis on their CRYPTO-1 algorithm implemented on the Mifare cards [10].

Reverse-Engineering Since NXP had not released any details on the card or the cryptographic implementation of their *CRYPTO-1* algorithm, Nohl's team had to reverse engineer the algorithm and implementation from the hardware. Most manufacturers deemed this process to be too costly and complicated for a simple attacker, however Nohl's team found a simple and inexpensive way to do this.

The team used acetone to dissolve the chips and then used a fine sandpaper to polish each layer of circuitry, taking multiple pictures of it through a standard microscope. By doing minor image post-processing the team managed to identify the components and reconstruct most of circuitry. Some of the missing pieces of information, such as the inputs and outputs of certain components were derived from the communication protocol of the Mifare card.

CRYPTO-1 Algorithm

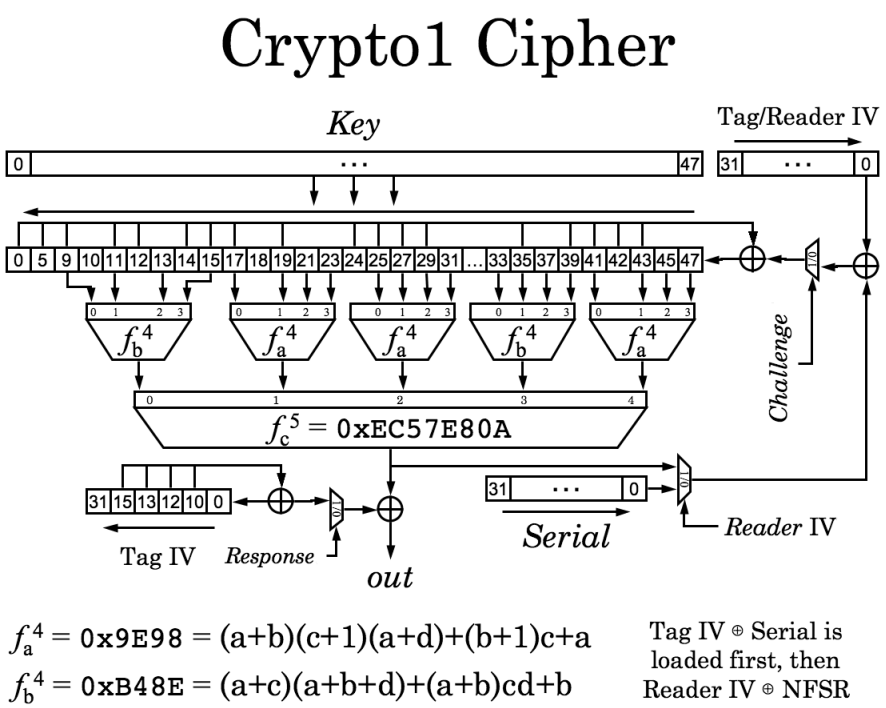

Figure 4: CRYPTO-1 Algorithm Overview

The CRYPTO-1 stream cipher consists of 48-bit linear feedback shift register (LFSR) and a filter function $f(\cdot)$, which itself conists on multiple implementations of smaller functions f_a & f_b. There is also an onboard *Random Number Generator* (RNG) consisting of a 16-bit LFSR.

Each time the RFID tag is powered, the main 48-bit LFSR is set to the secret key, and the RNG is set to a constant value (the same value on every power-up).

As soon as the device is powered up, the RNG starts producing values. The tag ID is then xored with a randomly generated number and shifted into the 48-bit LFSR. That same random number is sent to the reader as a first challenge where the reader has to prove its knowledge of the secret key. In each clock cycle the filter function will compute a bit of the stream using 20 LFSR bits, and then sent to the reader as part of the answer.

After both reader and tag have verified each others validity, communication then proceeds in a challenge-response manner, where the reader will send commands to the card and the card will respond with the appropriate

response. All of this communication is encrypted as aforementioned.

Cipher vulnerabilities

Key Length The first clear issue was the limited secret key size. A 48-bit secret key provides 2^{48} possible keys, which makes brute-forcing a viable solution. At the time, the team implemented a key cracker with 64 FPGAs running in parallel. With simply intercepting 2 challenge-response exchanges between a legitimate tag and reader, it could resolve the key in less than 50 minutes.

Random Number Generator Another critical issue is the RNG implementation. The RNG consists of a 16-bit LFSR which is initialized to the same value on each power-up. It is clocked at 106kHZ and cycles through all possible values in about 0.6 seconds.

Firstly, the LFSR utilized only produces 65,535 possible states, which is highly insufficient length for cryptographic application due to the lack of entropy. However the main issue is that each random value depends on the number of clock cycles elapsed since powering up. Furthermore, the designers enforced the RNG to initialized to the same constant value. This means that an attacker could control the timing of the protocol and know exactly what random number to expect.

A skilled attacker could have a precomputed cookbook which allows him to trim down the time needed to resolve the secret key of a card given just 2 challenge-response exchanges.

Biased Filter Functions Another issue presented in the Mifare card is that the filter functions happen to be statistically biased [9]. The researchers computed the bias by testing every possible key and counting how many 0's and 1's were outputted. The bias for each function were $f_c = 10/16$, $f_a = f_b = 5/8$. This vulnerability allows an attacker to discover 1 bit of the key at a time.

This attack consists on sending challenges to the reader and analyzing the first bit of the key stream sent in the response. Initially you hold the inputs for 1 of the $f_{a/b}$ functions equal, while changing other bits in each challenge. By seeing the impact that each new challenge has on the output one can statistically recover the bit of that $f_{a/b}$ block. One could then repeat the attack for successive blocks. Eventually one could recover all the bits, however, the researchers advised that some bits are considerably harder to

recover than others. Instead, this attack could be done for a couple of bits and then followed by the bute-forcing attack.

Attack Scenario A potentially cheap attack would consists of an attacker actively scanning a valid Mifare card, immediately obtaining the Tag ID[2]. Then the attacker could pretended to be a legitimate reader and, controlling the protocol timing, query the card. The responses could then be used on a precomputed rainbow table and within a few minutes the secret key would be known. At this point the attacker could choose to clone the card (e.i. effectively stealing the card) or change the data inside the card (e.g. adding fare money to a subway card)

3.2 DST Card

Introduction Another famous case of RFID reverse-engineering is Bono's cracking of the Texas Instruments' Digital Signature Transponder (DST) [3]. Even thought DST was broken in 2005, there are still more than 7 million transponders in use on the Exxon-Mobile Speedpass payment system. Furthermore some car manufacturers such as Ford, Toyota and Nissan still use these devices as theft-deterrent mechanism.

Reverse-Engineering As with the Mifare card, the DST system is proprietary and kept secret. The team of researchers found some rough schematics on the *DST40* cipher available on the Internet that gave a very rough idea of how the system worked. However, since the schematic lacked many details, the team decided to further analyze the DST by treating it as a *black box* and examining its logical outputs for chosen inputs. For their experiments they used the freely available TI Series 2000 - LF RFID Evaluation Kit which contained a reader, an antenna, and a variety of other RFID devices.

DST40 Cipher The DST40 cipher is essentially a 40-bit feedback shift register. Meaning, in each round of operations, inputs from the challenge are passed through a set of logical functions. This output is then fed back into the feedback register. The function F, ran on every round of operations, can be split into 3 layers of logical operation:

$\star$ The first layer consists of a set of 16 functions (f-boxes). The input for this functions alternates between 2 and 3 bits of the challenge.

[2]Recall that Tag IDs are always available for anyone to scan.

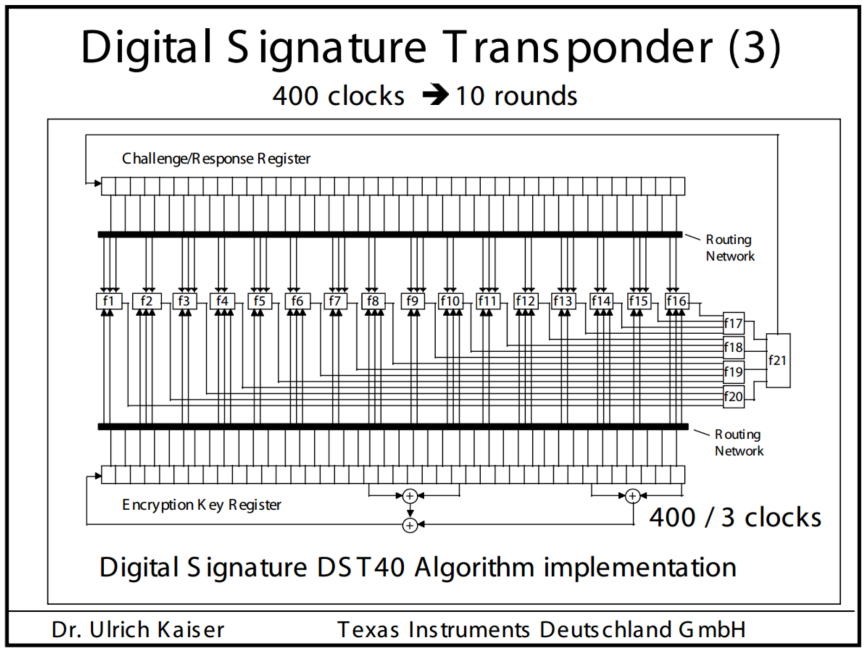

Figure 5: Original Schematic of Kaiser Cipher

⋆ The second layer consists on 4 functions (g-boxes) which feed from the output of the f-boxes.

⋆ The final layer consists of 1 function (h-box) which feeds from the g-boxes and outputs 2 bits, which is the final result of F.

Further investigation led the team to discover that the secret key was being shifted into the register every 3 cycles (starting with the second cycle) with function $k_0 = k_{39} \oplus k_{37} \oplus k_{20} \oplus k_{18}$.

The last challenging part was identifying the structure of the f,g and h-boxes. The team accomplished this by checking the changes in the output of F when flipping 1 bit of the input. Doing this for all possible inputs showed the correlation between inputs and outputs and allowed to distinguish the f-box, and subsequently the g-box and h-box.

DST protocol Once the cipher was discovered, the team looked at the interactions between the tag and a valid reader. After some experimentation the team placed all the pieces together and came up with the full DST protocol.

The DST protocol is a challenge-response mechanism where an initial

reader transmits a challenge request to a responding tag. The challenge sent by the reader consists of an 8-bit opcode and a 40-bit challenge. The opcode identifies the request type (action to be done). The tag responds to the reader with its 24-bit serial number and a 16-bit CRC to verify its correctness. The reader can then match and verify the validity of the tag.

Breaking the Keys Given a single exchange of information between a valid tag and reader, the researchers managed to compute the secret key. The team experimented with different devices and found that if they precomputed Hellman tables they were able to find the secret key with 99+% certainty in under 2 minutes [3].

Cipher Vulnerabilities

Key Length The main vulnerability lies on the small key space of the cipher. A 40-bit key is sufficiently small to brute-force. Using Hellman tables precomputations of just under 10GB one could find the secret key in a matter of minutes.

Secret CRC start value The addition of a cyclic redundancy check to the DST protocol was done as a measure of error checking and a way to add additional security. The problem is that the CRC starting value is the same for every single DST. Given a transmission with its companion CRC code, one could attempt to compute the CRC of the data with all 2^{16} possible CRC starting values in under a second. Once the CRC start value is found, one can use it for all other DSTs.

Attack Scenario An attacker near a gas station could potentially intercept the communications between SpeedPass cards and the pump readers. He could then find the secret keys of these cards and later spoof them to buy free gas.

4 Vulnerability Mitigation Strategies

Many researchers have focused on solving issues relating to the attacks and weakness explained above. We will now briefly look at some mitigation strategies to avoid or amend important vulnerabilities.

4.1 Mitigating Eavesdropping

Due to the wireless nature of RFID technology eavesdropping is unavoidable. However, there are precautions we can take to reduce read ranges and minimize tag exposures.

A simple precaution against active eavesdropping could be to keep tags under Faraday shielding when not in use. Many companies have developed metal enlaced wallets and other products to keep RFID-enabled credit cards and other keycards secure. However, this does not prevent passive scanning in any way.

One precaution against distanced active eavesdropping is having RFID tags aware of the distance between itself and the reader. Some authors have demonstrated the possibility of using signal-to-noise ratio to determine this distance [11]. While the method is not perfect, it could be used in systems such as keycards, where we expect the reader to be in close proximity to the tag and thus the tag could refuse to answer queries that come from distanced readers [3].

A precaution against passive eavesdropping could be adding metal shielding around RFID readers to limit the read range of any interaction. This is not suitable for every situation, but some car companies already add partial cylinders around their car's ignition slots to limit the read range between the car RFID reader and the car key [3].

4.2 Mitigating Relay Attacks

Relay attacks could be mitigated by adding distance bounding protocols to the RFID system. Bounding protocols are simple timing mechanisms to test the distance between a tag and a reader. Since the original tag and reader are potentially separated by substantial distances, the interaction between them could be slower than expected, and thus an attack could be discovered and stopped. This is still an area of research and there are many possible approaches to these protocols.

One approach described by Hancke and Kuhn [12] is to provide a series of challenge that the reader must process and respond to within a fixed time. To avoid pre-computation of challenge-response tables, the reader provides one nonce [3] which will be used in the challenge calculations. The acceptable time by which the tag must respond has to be fine tuned to allow physically present tags to pass the tests, but relay attacks (which most likely run over a network) to fail.

[3]Nonce: a random number only used once.

4.3 Mitigating Authentication Vulnerabilities

The problem of authentication is an umbrella issue that if properly address, could solve *Man in the Middle* attacks, *tag cloning*, and *spoofing*.

4.3.1 Secure Forward Link

Tag Authentication One method proposed for secure authentication relies on each tag containing a Physically Unclonable Function [13]. PUFs are hardware challenge-response mechanisms that rely on hardware intricacies and micro-variations to produce results unique to each device. PUFs aim to provide reproducible responses to any given challenge for a particular hardware device. The readers will be connected to a database that has a set of challenges specific for every set of PUFs, allowing it to properly challenge each tag.

When a reader queries a tag, the tag will respond with it's tag ID. The reader will select the proper challenge from the challenge set to send to the tag. Upon transmission, the tag will use these challenges on it's PUF and generate a key K. This key is fed into a stream cipher which will produce a another key K_s. The original key K is sent to the reader and the reader will use it along with the challenge set to calculate Ks. Once both have K_s they can communicate securely.

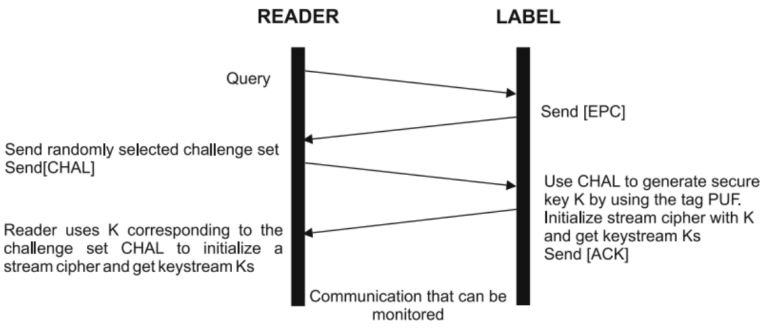

Figure 6: Protocol for achieving a secure communication channel.

Mutual Authentication Notice the above method only assures the reader that any given tag is valid. However, we can enhance this method to provide mutual authentication of tag and reader.

Following the previous example, once a reader has queried the tag and the secret key K_s has been calculated, the reader will send a random number

$R_n \oplus K_s$ to the tag. The tag will initialize the stream cipher with R_n, calculate the subsequent K_t, and send $K_s \oplus K_t[0]$. The reader will use $K_t[0]$ to authenticate the reader and then send $K_s[1] \oplus K_s[4]$. Finally the tag will use $K_s[4]$ to verify the reader.

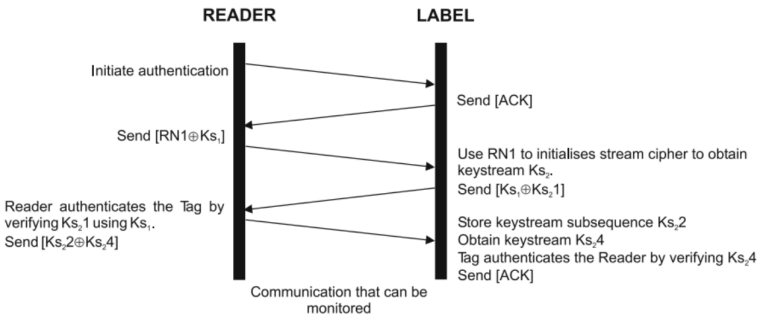

Figure 7: Protocol for mutual authentication (assuming initial authentication was established).

4.4 Mitigating RNGs Vulnerabilities

Random Number Generators (RNGs) are a critical part of any cryptosystem. One of the major challenges of implementing RNGs in RFID devices relies on the limited power supply and circuit space provided. This has lead some manufacturers to integrate sub-par implementations, such as LFSRs, which produce low-entropy randomness. These faulty implementations, similar to the ones found in DST and Mifare make it easier for attackers to brute-force cryptographic ciphers.

Source of entropy Most of the implementations proposed to fix RNGs in RFID focus on the importance of having a highly entropic source of randomness. A few possible approaches to accomplish this could be:

⋆ **Direct Amplification** This approach focuses on using a high-gain high-bandwidth amplifier to amplify the AC voltage produced by a noise source [14]. This noise source can be thermal or simply electronic shot noise. While this is a simple and small implementation, it does require some level of shielding, or otherwise it could be susceptible to side-channel attacks where attackers introduce specific noise in the system.

⋆ **Oscillator-based TRNG** Another possible approach consists having a jittered low-frequency clock sample a high-frequency clock [15]. This model's timing can be enhanced by feeding the output of this TRNG into a modified 16 bit LFSR. If we assume proper initial seeding, then we will be able to produce 16 bits of random numbers in the time it took our original TRNG to produce 1 bit.

⋆ **PUFs** A rather unconventional proposition is to use Physically Unclonable Functions (PUFs) as a source of randomness. Remember that PUFs are hardware challenge-response mechanisms that rely on hardware intricacies and micro-variations to produce results unique to each device. Under normal circumstances one would want to use PUFs that provide reproducible response to any given challenge for a particular hardware device. However there are some *bad* PUFs which give unpredictable answers. These PUFs might be useful for random number generation.

Silicon delay-based PUFs are a particularly simple type of PUF that produces unpredictable results. These consist of a chain of interconnected challenge gates connected at the end to a simple D-Latch. These PUFs tend to have metastable conditions where the answer to any given challenge can be 0 or 1 with equal probability.

O'Donnel, Suh and Devadas [16] suggest that an RNG could be built by having a series of PUFs running different challenges. Their output is checked by a Von Neumann corrector (to eliminate any bias) and finally feed back as an input to the PUFs themselves.

Power-on Generation Many of the implementations listed before consume a considerable amount of power which might make their integration into RFID more complicated. If one is willing to sacrifice response time, a possible solution [15] could be to place the device in *"number generation mode"* as soon as the tag gets powered while keeping the rest of the circuit blocks in *"sleep mode"*. After sufficient random numbers are generated, the RNG is turned off and the rest of the circuitry is woken up. This allows us to feed both the RNG and the rest of the circuit with a larger power budget.

4.5 Open Security Standards

While this is not a particular technical flaw, it is pertinent to point out how the RFID industry has relied heavily on the idea of *security by obscurity*. In fact, most of the devices currently in the market work with proprietary cryptographic protocols which have been purposely kept secret. In fact,

some companies have gone as far as suing researchers that publish papers on their findings [10], and, in a more comical incident, even forced the popular *Myth Busters* show from airing an episode on RFID hacking.[17].

It has been shown multiple times that this practice leads to a false sense of security. It is better to rely on standard, publicly reviewed ciphers than on obscure proprietary ones. In fact many authors have proposed implementations of peer-reviewed ciphers specifically crafted for RFID, such as 128-bit AES [18], HMAC-SHA1 [19], HB^+ [20], among others.

Another flawed assumption that companies seem to take is that reverse-engineering hardware is expensive and complex for any attacker. As we have shown on the previous study cases, such reverse-engineering is viable with inexpensive materials.

In the end, relying on *security by obscurity* will most likely lead to breaches, merchandise recall, and lost profits.

5 Conclusion

RFID has come a long way since we first saw its integration onto consumer products, however there is still more research to be done to maintain the privacy and security of its users. With the rise of *Internet of Things* (IoT) devices and smart appliances, we will probably encounter RFID in more situations. This paper has given a concise look at the overall implementation and issues of cryptographic protocols within the context of passive RFID devices. Hopefully this information can provide interested researchers a good starting point in the field of passive RFID cryptography.

References

[1] J. Bond, "Rfid market to exceed $10 billion in 2015," 2015.

[2] C. D. Ranasinghe and H. P. Cole, *Networked RFID Systems and Lightweight Cryptography: Raising Barriers to Product Counterfeiting*, ch. An Evaluation Framework, pp. 157–167. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[3] S. C. Bono, M. Green, A. Stubblefield, A. Juels, A. D. Rubin, and M. Szydlo, "Security analysis of a cryptographically-enabled rfid device," in *Proceedings of the 14th Conference on USENIX Security Symposium - Volume 14*, SSYM'05, (Berkeley, CA, USA), pp. 1–1, USENIX Association, 2005.

[4] F. Brown and B. Fox, *RFID Hacking*. DEFCON 21, Aug 2013.

[5] Y. Oren and A. Shamir, *Remote Power Analysis of RFID Tags*. Weizmann Institute of Science, Aug 2006.

[6] P. Paganini, *Near Field Communication (NFC) Technology, Vulnearbilities and Pricipal Attack Schema*. INFOSEC Institute, June 2013.

[7] Marc, *https://hackaday.com/2008/06/09/rfid-reader-denial-of-service/ (Accessed: 20 Aug 2016)*. Hackaday.com, CanSecWest, 2008.

[8] K. Nohl, D. Evans, S. Starbug, and H. Plötz, "Reverse-engineering a cryptographic rfid tag," in *Proceedings of the 17th Conference on Security Symposium*, SS'08, (Berkeley, CA, USA), pp. 185–193, USENIX Association, 2008.

[9] K. Nohl, *Cryptoanalysis of Crypto-1*. University Of Virginia, 2008.

[10] E. Mills, *Dutch chipmaker sues to silence security researchers*. cnet.com, Jul 2008.

[11] K. Fishin, S. Roy, B. Jiang, K. P. Fishkin, S. Roy, and B. Jiang, "Some methods for privacy in rfid communication," 2004.

[12] G. P. Hancke and M. G. Kuhn, "An rfid distance bounding protocol," in *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*, pp. 67–73, IEEE, 2005.

[13] D. C. Ranasinghe, "Lightweight cryptography for low cost rfid," in *Networked RFID Systems and Lightweight Cryptography*, pp. 311–346, Springer, 2008.

[14] C. S. Petrie and A. Connelly, "A noise-based ic random number generator for applications in cryptography," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 47, no. 5, pp. 615–621, 2000.

[15] W. Che, H. Deng, W. Tan, and J. Wang, "A random number generator for application in rfid tags," in *Networked RFID systems and lightweight cryptography*, pp. 279–287, Springer, 2008.

[16] C. W. Odonnell, G. E. Suh, and S. Devadas, "Puf-based random number generation," *In MIT CSAIL CSG Technical Memo*, vol. 481, 2004.

[17] N. Patel, "Mythbusters rfid hacking episdoe canned by credit card company lawyers," 2008.

[18] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong authentication for rfid systems using the aes algorithm," in *Cryptographic Hardware and Embedded Systems-CHES 2004*, pp. 357–370, Springer, 2004.

[19] H. Krawczyk, R. Canetti, and M. Bellare, "Hmac: Keyed-hashing for message authentication," 1997.

[20] A. Juels and S. A. Weis, "Authenticating pervasive devices with human protocols," in *Advances in Cryptology–CRYPTO 2005*, pp. 293–308, Springer, 2005.

[21] G. de Koning Gans, J.-H. Hoepman, and F. D. Garcia, *A practical attack on the MIFARE Classic*. Springer, 2008.

[22] C.-L. Chen and C.-Y. Wu, "An rfid system yoking-proof protocol conforming to epcglobal c1g2 standards," *Security and Communication Networks*, vol. 7, no. 12, pp. 2527–2541, 2014.

[23] M. H. Habibi, M. Gardeshi, and M. R. Alaghband, "Practical attacks on a rfid authentication protocol conforming to epc c-1 g-2 standard," *arXiv preprint arXiv:1102.0763*, 2011.

[24] M. Burmester and B. De Medeiros, "Rfid security: attacks, countermeasures and challenges," in *The 5th RFID Academic Convocation, The RFID Journal Conference*, 2007.

[25] A. Juels, "Rfid security and privacy: A research survey," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 2, pp. 381–394, 2006.

[26] G. P. Hancke, "Practical eavesdropping and skimming attacks on high-frequency rfid tokens," *Journal of Computer Security*, vol. 19, no. 2, pp. 259–288, 2011.

[27] G. Hancke *et al.*, "Eavesdropping attacks on high-frequency rfid tokens," in *4th Workshop on RFID Security (RFIDSec)*, pp. 100–113, 2008.

[28] D. R. Thompson, N. Chaudhry, C. W. Thompson, *et al.*, "Rfid security threat model," in *Conf. on Applied Research in Information Technology*, 2006.

[29] L. Grunwald, "New attacks against rfid-systems," *GmbH Germany*, 2006.

[30] R. W. Schreur, P. Van Rossum, F. Garcia, W. Teepe, B. Jacobs, G. D. K. Gans, R. Verdult, R. Muijrers, R. Kali, and V. Kali, "Security flaw in mifare classic," 2008.

[31] M. Almeida, "Hacking mifare classic cards," in *Black Hat Regional Summit Sao Paolo*, 2014.