Feasibility of the RFID Guardian as a relay attack platform*

Dennis Andriesse – da.andriesse@few.vu.nl

Abstract

RFID technology is being introduced in increasingly many sensitive environments. To implement appropriate security measures in important RFID systems, we must understand how these systems can be exploited.

In this paper, we analyze the feasibility of using RFID Guardian devices to mount a relay attack against ISO 14443A and ISO 15693 compliant RFID systems.

In order to get an accurate impression of the RFID Guardian's suitability as a relay attack platform, we first perform a theoretical feasibility study. Next, we analyze the ISO 14443A and ISO 15693 RFID standards to identify vulnerabilities which can be exploited in a relay attack. Finally, we implement and test our findings in a simulated environment.

1 Introduction

RFID is all around us. Stores use RFID tags to label products and protect them against theft. Modern public transportation tickets contain RFID tags. Even in extremely sensitive applications like access passes for buildings and passports, RFID tags are used.

Despite the many areas where they are applied, RFID tags are simple devices. They are small chips which are capable of sending and receiving radio messages. Using these radio messages, RFID tags can identify whatever it is they are attached to.

RFID tags are used in conjunction with RFID readers. These readers periodically broadcast a radio message, looking for nearby tags. Tags which receive such a message send back a response to the reader. The reader selects one of the nearby tags to continue the conversation with, and the rest of the tags are excluded from communication until the reader is done talking to the selected tag. The phase in which a single tag is selected from all the available tags is called the *anticollision phase*. RFID readers also usually supply tags with power and a reference clock signal. Tags which have their own power source and clock do exist, but are not used with the protocols we discuss in this paper.

RFID systems can make our lives easier. There are drawbacks, however. An insecure tag in an access pass for a government building or a high–tech company's R&D department is a serious security risk. An insufficiently secured electronic passport can even facilitate identity theft.

Additionally, RFID tags can threaten an individual's privacy because they contain a unique identifier, and will tell it to any reader which requests it. This means that an individual carrying one or more tags can be tracked by any nearby RFID reader.

The RFID Guardian is a portable device which protects an individual's RFID tags and privacy by blocking read requests from unauthorized readers. It has an analog front–end, which contains an RFID transceiver and can be used to receive RFID frames, send out jamming signals, and even send spoofed frames [RCT05]. Besides using the RFID Guardian to manage tags, it is also possible to perform security and penetration testing with it. This is how we apply the RFID Guardian in this paper.

Instead of going through much trouble to break a system's actual security layers, a relay attack circumvents them. In a relay attack, we don't try to clone a tag or make a fake tag to fool the target system. It is much easier to simply *relay* a reader's requests to an actual tag, which knows how to respond, and then relay this legitimate response back to the reader. We don't need to know anything about things like the system's encryption algorithm, or whether it contains programming errors. The following example illustrates this idea.

*Bachelor's thesis, June 2010. Supervisor: Melanie Rieback (melanie@cs.vu.nl). Second reader: Rutger Hofman (rutger@cs.vu.nl).

Two corporate spies, Trudy and Mallory, want to enter a high–tech company's office. The door of the office is secured by an RFID system, and to open it they need an RFID tagged access pass. Rather than trying to fake or steal a pass, they decide to execute a relay attack.

Alice, an actual employee of the company, is sitting in front of the building having lunch. Trudy and Mallory know Alice must have an access pass with her. So Mallory sits next to Alice and takes out a sandwich too, while Trudy stands by the door. Both Trudy and Mallory have special RFID transceivers in their bags.

When the RFID reader at the door transmits a request looking for tags, it is forwarded by Trudy's transceiver to Mallory's transceiver. Without Alice knowing a thing, her tag answers the request which is now coming from Mallory's transceiver. Mallory's transceiver forwards this response back to Trudy, and Trudy's transceiver sends it to the reader at the door. Trudy now has access to the building.

In this example, Trudy's transceiver essentially acted as an "extension" of Alice's tag from the viewpoint of the door's reader, and Mallory's transceiver acted as an "extension" of the reader from the viewpoint of Alice's access pass. The primary goal of this paper is to determine the feasibility of using RFID Guardian devices in the place of these transceivers.

The rest of this paper is outlined as follows. In section 2, we introduce related work. In section 3, we perform a theoretical feasibility study to determine what the difficulties in an actual implementation of the attack would be. Next, in section 4, we try to find vulnerabilities in the ISO 14443A and ISO 15693 RFID protocols which make launching a succesful relay attack easier. In section 5, we describe the results of our simulated relay attack implementation. Finally, in sections 6 and 7, we discuss our findings, and conclude the paper.

2 Related Work

The RFID Guardian is thoroughly introduced in [RCT05] and [RGC+06].

An in–depth introduction to RFID technology in general can be found in [AI08].

An overview of much relevant research into relay attacks can be found in [HMM09].

Systems shown to be vulnerable to relay attacks include: the Czech e–passport [HR07], a real–world RFID based e–voting system [OW10], and RFID–based payment systems [KW05] .

Two of the most commonly used RFID standards are ISO 14443 [ISO01] and ISO 15693 [ISO00]. Layers of ISO 14443 also sometimes serve as a base for other standards. For example, Mifare is built on top of ISO 14443A layer 3, and Desfire is built on top of ISO 14443A layer 4.

Distance bounding protocols provide a potential layer of defense against relay attacks, although they are still highly experimental and not employed in any real RFID systems today [DM07].

3 Theoretical feasibility

In this section we evaluate the theoretical feasibility of mounting an RFID relay attack with the RFID Guardian.

Because the RFID Guardian is in effect a portable computer running a real–time operating system, it should in principle be flexible enough to act as a relay device. However, two important issues are critical to the success of the attack:

- the device must have a suitable relay channel with sufficient range; and
- the delays introduced by the device and the relay channel must be sufficiently small so that they do not upset communication between the real reader and tag.

These two issues are discussed in detail in the following subsections. In our discussion, we will refer to the real reader simply as *the reader*, and to the real tag as *the tag*. The "extended" reader and tag implemented on the RFID Guardian devices will be referred to as the *proxy reader* and *proxy tag*, respectively.

3.1 Relay channels

The RFID Guardian offers a variety of communication channels. As of version 4, these include:

- a class 1 Bluetooth interface,

- a 10/100 Mbit/s Ethernet port,
- a High Speed USB port,
- a user replaceable radio interface.

In this section, we discuss which of these channels are suitable as relay channels.

3.1.1 Bluetooth

Class 1 Bluetooth devices have a nominal range of about 100 meters, which should be sufficient for a relay attack. Unfortunately, Bluetooth packet exchange is based on a clock which ticks at 312.5 μs intervals, and this adds so much latency to the Bluetooth channel that by default it is not practical as a relay channel [Blu09]. It may be possible to increase the Bluetooth clock speed, if the hardware allows it. It could be worthwhile to investigate this, but such an investigation is beyond the scope of this project.

3.1.2 Ethernet

The Ethernet interface is unpractical in many cases, because it is wired. However, in cases where a wired relay channel is acceptable Ethernet may be a good option. It offers sufficient bandwidth, and if there is little contention on the channel latency is likely to be sufficiently low.

It is also possible to connect a gateway to the Ethernet port which forwards the Ethernet communications wirelessly, but this will introduce additional latency to the setup. Furthermore, Ethernet does not supply any power to devices, so a gateway will need to be connected to an external power supply. Even if this is a battery, the relay setup will become bigger and more difficult to hide.

3.1.3 USB

USB is also wired, but wireless communication devices for USB are readily available, and are generally quite small. Such a device may even be able to draw its power from the USB port, eliminating the need for an external power supply. USB's isochronous ("real–time") mode may be suitable for a relay channel. This would require that the mandatory buffering and unbuffering of USB packets is done fast enough, however. An isochronous transfer is guaranteed a certain amount of time per USB frame. Thus, isochronous mode provides bounded latency. Every frame carries some USB control information, so for every send there may be a delay waiting for the control data to finish [USB00]. If a sufficient amount of frame–time can be reserved, and buffering/unbuffering can be done fast enough, USB may be a feasible relay channel.

3.1.4 Radio

Protecting an individual's RFID tags —which is the original goal of the RFID Guardian— requires a range of only one to two meters for the radio transceiver [RGC+06]. Such a short range is usually insufficient for a relay attack. To allow a practical relay attack, the transceiver should offer a range in the order of at least 50 meters. Thus, if current analog front–ends for the RFID Guardian do not offer sufficient range, a new one will need to be designed. As the RFID Guardian has a modular design, it should not be a problem to replace the analog front–end.

Another possible problem is that the currently available radio frontend for the RFID Guardian has only one antenna. This means that we will either need to receive an entire RFID frame before relaying it, or use a channel other than the radio interface for communication between the two proxy Guardians. It is also possible to design a radio frontend with two antennas. If the two antennas operate at different frequencies, we can relay RFID communications in real time. Relaying in real time allows for smaller delays, but also makes any active modification of the data stream more difficult.

The radio transceiver conforms to the same timing demands that the RFID devices have to conform to. Thus, the radio channel itself should not introduce any latency problems.

3.2 Timing constraints

RFID standards usually impose fairly stringent timing constraints on communication between tags and readers. A relayed response might be rejected by the reader if it arrives too late. Therefore, it is important to keep delays in the relay setup as small as possible. In this section, we investigate the delay a relay setup using two RFID Guardians is likely to introduce, and how this delay will affect the relay attack.

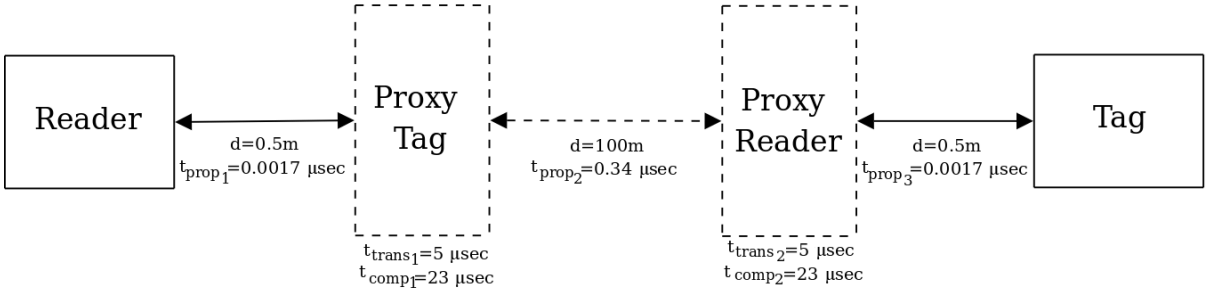

Figure 1: An example relay setup and the delays it will produce.

3.2.1 Relay setup delays

The most important delays a relay setup will encounter are illustrated in figure 1. In the illustration, we assume that the proxy reader and proxy tag are implemented on RFID Guardians. Also, we assume that the relay channel has sufficient bandwidth and sufficiently low latency, such that the only extra delay introduced by the relay channel is an additional propagation delay. We calculate propagation delays under the assumption that the signals travel at the speed of light.

The distances between the various devices in the illustration may differ in a real relay attack. We have chosen reasonable estimates, which should be representative of most real relay attacks.

Abbreviations used in figure 1 are explained in table 1. t_{trans} and t_{comp} are due to [RGC+06], and t_{trans} assumes that the RFID Guardians use a radio interface for the relay channel. t_{comp} is a worst case estimate.

Symbol	Meaning
d	Distance
t_{prop}	Propagation delay
t_{trans}	Transmitter startup time
t_{comp}	Guardian computation delay

Table 1: Abbreviations used to denote delays.

A request from the reader to the tag must cross through the entire relay setup, and the response from the tag must cross through the entire setup again. Thus, the worst case total delay introduced by the relay setup for a single tag response is as follows.

$$2 \times (\sum_{i=1}^{3} t_{prop_i} + \sum_{i=1}^{2} t_{trans_i} + \sum_{i=1}^{2} t_{comp_i}) \approx 112.7\mu s$$

The processing time needed by the relay program running on the proxy reader and proxy tag is not included in the picture. Given that the relay program does not need to be very complex, the computation delay it introduces should be low.

It should be possible to keep the radio transmitters active after forwarding a request, in expectation of the response. Doing so will reduce the delay by 10 μs. If we start up the transmitters before the actual attack, and keep them active throughout the entire attack, we can eliminate t_{trans} entirely, thus saving 20 μs in total.

It is clear that t_{comp} is by far the greatest delay. t_{comp} is actually composed of several delays. One significant factor is the delay caused by receiving RFID frames entirely before forwarding them.

It should be possible to lower t_{comp} significantly by forwarding frames in real time, or at least per byte. This is already possible in software, but real time forwarding introduces multiplexing problems which raise the need for either a radio frontend with two antennas, or a relay channel other than the radio interface.

As an aside, in [HK08], Hancke et al. propose sending a faster than normal clock signal from the proxy reader to the tag, effectively "overclocking" the tag. This may cause the tag to process requests faster than usual, and also send back its response at

a higher clock rate. This would grant the attacker more leeway for forwarding the tag's response.

It is probably possible to implement this idea on the RFID Guardian, but it does require a radio interface which supports variable frequencies. Such a radio interface is currently being designed for the RFID Guardian, so we expect that tag overclocking will be possible on the RFID Guardian in the near future.

We estimate that it should be possible to reduce our worst case response delay by at least 30 μs, by eliminating t_{trans} and implementing some additional optimization, such as forwarding frames in real time or overclocking the victim tag. This would bring the delay caused by the relay setup down to 82.7 μs or less.

3.2.2 ISO 14443A timings

ISO 14443A specifies its most stringent timing constraints during the anticollision phase. During this phase, a Frame Delay Time is used which specifies the time between two frames transmitted in opposite direction. The FDT is mainly needed to ensure that tags are transmitting synchronously, so that the reader can reliably detect collisions caused by tags transmitting their Unique ID's simultaneously during anticollision.

During anticollision, ISO 14443A–3 requires an FDT of 86.43 μs if the last bit sent by the reader was a 0, and 91.15 μs if the last bit sent by the reader was a 1. This means that, according to the standard, tags should start sending their responses after exactly this delay.

At first sight this seems like a problem if we can not significantly reduce the delay introduced by our relay setup. However, Hancke et al. report that real readers rarely enforce the FDT constraints strictly [Han05]. Instead, they often accept delayed anticollision responses as long as there are no other tags sending misaligned frames at the same time. That is to say: in a real scenario synchronization seems to be maintained because the tags comply to the required timing, not because it is strictly enforced by the reader. This means that a relay attack is still likely to work, as long as no other tags are within range of the reader during the attack. It does not matter if other tags are in the range of the proxy reader, because their responses will be aligned.

After anticollision is finished, ISO 14443A timings tend to be more lenient, as described next.

ISO 14443A–4 specifies the Frame Waiting Time as the time within which a tag must answer after the end of a request from a reader. Its value is defined in the standard as at least 302 μs, but the default value for the FWT is about 4.8 ms. This should be more than enough time for any relay attack.

Mifare does not specify any timings, but instead uses the ISO 14443A–3 timings. After the anticollision sequence, ISO 14443A–3 requires responses to be aligned according to the formula

$$(n \times 128 + 84)/f_c \tag{1}$$

if the last bit in the request is a 1, or

$$(n \times 128 + 20)/f_c \tag{2}$$

if the last bit is a 0, with $n \geq 9$ and $f_c = 13.56$ Mhz. When $n = 9$, timings are identical to those during anticollision. Aligning responses correctly should not be a problem.

3.2.3 ISO 15693 timings

In ISO 15693, a tag must wait for at least 318.6 μs before sending a response to any request. The maximum waiting time before sending a response is 323.3 μs. This means that an ISO 15693 tag will not start responding before these timeouts have expired, and there is very little time left to relay a tag's response. Consequently, a response from a relayed tag will almost inevitably be late from the viewpoint of the reader. The reader may or may not still accept the response. This will need to be determined experimentally, but we suspect that most readers will not accept frames after the kind of delay introduced by our relay setup.

It should be noted that it may be possible to gain time by overclocking the tag. The combination of a sufficiently low delay in the relay setup and a sufficiently high overclock of the tag might be enough to make a relay attack against an ISO 15693 system possible after all.

3.3 Findings

From our theoretical study, we conclude that the RFID Guardian offers several feasible relay chan-

nels, and that the most practical candidate is probably the radio frontend.

We conclude that the RFID Guardian appears to be a feasible platform for a relay attack. We expect no problems relaying ISO 14443A–4, Desfire, or Mifare communications, but suspect that a relay attack on ISO 15693 will fail unless we identify vulnerabilities in the protocol itself, or succeed in sufficiently overclocking the victim tag.

4 Vulnerabilities

In this section, we identify a number of vulnerabilities which may be present in RFID systems. These vulnerabilities can potentially be exploited to increase the likelihood of success in a relay attack.

4.1 Generic RFID vulnerabilities

4.1.1 Offline anticollision

A possible way to get around strict anticollision timings, is to pre–record the information required for an anticollision round (like a tag's Unique ID), and during the relay attack use this information to fake an anticollision sequence. As tags will transmit their Unique ID to any reader which requests it, it should be easy to obtain.

In this scenario, the proxy tag uses the pre–recorded information to perform an anticollision sequence directly with the reader, without forwarding any frames. In the RFID standards we analyzed, no encryption is used yet during anticollision, so this should be possible. At the same time, the proxy reader performs an anticollision sequence with the real tag, so the tag will be ready to receive frames from the real reader later. Thus, the reader thinks it has selected the real tag, and the tag thinks it has been selected by the real reader. The advantage here is that no relaying is required, so there is no abnormal delay. Once anticollision is complete, frames can be forwarded as usual [HR07].

4.1.2 Tag overclocking

As mentioned in section 3.2.1, it is possible to send a faster than normal clock signal to a tag in order to get it to process data and send a response more quickly. This could grant an attacker valuable extra time to relay a tag response back to the reader.

In Hancke's results, tag responses started about 10 μs sooner, and ended about 30 μs sooner for a 2 Mhz increase in frequency. With an overclock of 3 Mhz or higher, ISO 14443A responses appear to get too distorted to be useful [HK08]. Nevertheless, higher overclocks may be possible against other protocols, depending on the modulation technique used in the protocol.

4.1.3 SOF spoofing

In one of the RFID Guardian simulated readers, we encountered a vulnerability which grants unlimited time to relay a frame. The reader implements a timeout only while waiting for a Start of Frame marker to come in. Once it has received an SOF marker, it waits indefinitely for the rest of the frame.

This can be exploited quite easily by sending an SOF to the reader as soon as its request is received by the proxy tag, meanwhile relaying the request to the real tag. Once the response from the real tag is available, the rest of the frame can be sent to the reader.

We do not know if this vulnerability is present in any real readers. We suspect that most readers are immune to this exploit. Nevertheless, this is a potential vulnerability which can have serious consequences if it is present in a real reader.

4.2 ISO 14443A vulnerabilities

4.2.1 FWI spoofing

ISO 14443A–4 contains a feature which is especially interesting in the context of a relay attack. Once an ISO 14443A–4 compliant reader has selected a compatible tag, a number of parameters are set up for the rest of the conversation. Specifically, the tag sends a frame called an ATS (Answer to Select) to the reader. One of the fields contained in this frame is called FWI (Frame Waiting Time Integer). The Frame Waiting Time is the time within which a tag must start sending its response after the end of a frame from the reader. The reader will use any valid value the tag sends it. Since the ATS is an unencrypted frame, we can modify it any way we want! Moreover, the maximum value allowed for the FWT is 4949 ms, or almost 5 seconds. Thus, implementing a relay attack against ISO 14443A–4 should be trivial using this vulnerability.

Note that this is a layer 4 specific vulnerability. It will not work against Mifare, which is based on layer 3 of ISO 14443A. It should work against Desfire, which is based on layer 4.

4.2.2 S(WTX) spoofing

An ISO 14443A–4 tag might sometimes need more than the selected Frame Waiting Time to respond to a request from the reader. To accomodate this, ISO 14443A–4 specifies a special kind of frame, called an S(WTX) (Supervisory Waiting Time eXtension) frame. A tag can send such a frame to the reader to request more time, up to the maximum FWT of 4949 ms.

Of course, a proxy tag can spoof an S(WTX) frame just as well when more time is needed to relay a tag response. The temporary FWT determined by an S(WTX) takes effect for the duration of one tag response, so a new S(WTX) request will need to be sent from the proxy tag every time more leeway is needed for relaying a response.

S(WTX) spoofing provides an alternative to FWI spoofing, described in section 4.2.1, and gives us the same advantages FWI spoofing does. S(WTX) spoofing, like FWI spoofing, is a layer 4 specific vulnerability.

4.3 ISO 15693 vulnerabilities

4.3.1 Data rate flag spoofing

ISO 15693 supports both high and low data rates for communication between the reader and the tag. If the reader uses a low data rate, a marginal amount of time can be gained by telling the tag to send at high data rate. That is to say, if the tag sends at high data rate, the response should arrive marginally sooner than the reader using low data rate expects. If the reader already uses high data rate, there is no time to be gained.

Getting the tag to send faster can be achieved by setting a field called data_rate_flag in the reader's request to 1, meaning high data rate, before forwarding it to the tag. By itself, this is unlikely to make much of a difference, but it might provide an extra edge when used in combination with tag overclocking.

5 Simulator experiments

In order to test the findings from our theoretical feasibility study and vulnerability analysis, we have implemented a relay program in a simulated environment. The reason we used a simulated environment is that physical RFID Guardians were not available for testing within the time set out for this study.

The only major problem we encountered with the simulated environment is that timings are less predictable than in an actual RFID system. The simulations were executed in a desktop Linux environment. In such an environment, unexpected delays can occur at all times, for example due to process switches. We have been unable to fully eliminate these effects from our experiments. Despite this, we believe our tests give a good impression of the practical feasibility of our ideas, and that our results certainly warrant future testing on real RFID Guardians.

Simulated ISO 14443A/Mifare and ISO 15693 readers and tags were already available as part of the RFID Guardian project software. We built our relay program on top of the existing RFID Guardian library, and designed it to work with the already existing readers and tags. The relay program supports adding delays, in order to simulate the delay a real relay setup would introduce. It also implements some of the vulnerabilities mentioned in section 4, including tag overclocking. The entire program code is available from [SVN].

5.1 Findings

In this section we describe the results of our simulator experiments. The experiments were conducted on a quad–core AMD Phenom system running Ubuntu Linux. In all tests, the reader, tag and relay processes were each assigned a separate CPU core and given high scheduling priority.

5.1.1 Timing

To ensure that simulated tags and readers will work even if a time consuming process switch occurs, the RFID Guardian software implements a special leeway time when compiled for a simulator platform. Simulator programs add this leeway time to their usual timeouts, so that a process switch will not

cause a tag response to be rejected. The default leeway time in the order of 100 ms stands in the way of realistic experiments, because it means our relay program has much more time to relay responses than in a real scenario. For this reason, we also performed tests with the leeway time set to zero.

For ISO 14443A/Mifare, relaying did not work with the leeway time set to zero. This seems to be because the simulated reader uses very strict timings: it uses the minimal $n = 9$ in formula 1 and 2 for every frame. As stated in section 3.2.2, real readers are rarely this strict. They usually accept anticollision responses which are somewhat delayed, as long as there are no clashes with other tag's responses. Additionally, ISO 14443A–3 only requires tag responses after anticollision to be aligned according to some $n \geq 9$, and not to any fixed n. Strict anticollision timings could be circumvented using offline anticollision, as mentioned in section 4.1.1. Therefore, despite the behaviour of the simulated reader, we see no reason to assume that relaying ISO 14443A/Mifare will not work in real–life. As an aside, setting somewhat higher values for n in the simulator did allow the relay attack to work.

Relaying ISO 15693 without extra simulator leeway and without relay delay worked as long as no unexpected events like process switches occurred. However, as expected, added relay delay quickly reduced the rate of succesfully accepted frames. Even for a 10 μs delay, much shorter than the delay of a real relay setup, frames were only accepted by the reader sporadically. This seems to confirm our suspicion that ISO 15693 is quite intolerant to delays.

5.1.2 Implemented vulnerabilities

We implemented and tested several of the vulnerabilities mentioned in section 4. We did not implement offline anticollision due to the timing constraints of this project. However, the concept of offline anticollision is explored in more depth in [HR07]. Although FWI spoofing and S(WTX) spoofing seem very promising, we chose not to implement them because no well–tested ISO 14443A–4 simulator was available at the time of this study.

Although the susceptibility of real readers to SOF spoofing remains to be determined, our tests show that the concept is at least sound. We implemented this potential exploit, and used it succesfully against the simulated ISO 14443A/Mifare reader. In our tests, a valid SOF marker was sent back to the reader as soon as its request was received at the proxy tag. The reader accepted the SOF, and then commenced waiting for the rest of the frame, giving the relay program ample time to forward the tag's response.

ISO 15693 data_rate_flag spoofing was also implemented in the relay program. In our tests for this exploit, we set the simulated ISO 15693 reader to operate at low data rate, and instructed the relay program to communicate with the tag at high data rate. Unfortunately, there is no notable difference in the response times we measured with and without data_rate_flag spoofing against a reader using low data rate. Closer inspection of the simulated tag's source code suggests that it always sends at a constant data rate, even though the simulated reader supports both high and low data rates. This means that the usefulness of ISO 15693 data_rate_flag spoofing remains to be determined in real–life tests.

Our tag overclocking results were encouraging. We compared tag response times for no overclock, a 1 Mhz overclock, a 2 Mhz overclock, and a 3 Mhz overclock. For each overclock value, an average was taken over 10 samples, where extreme outlier values due to unexpected system delays were left out of the calculation so as not to skew the results. For ISO 14443A, we measured the time taken by an ATQA (Answer to Request, type A) frame to return in response to a REQA (Request command, type A) frame. For ISO 15693, we measured times for single–slot inventory responses. The test results are summarized in table 2.

% speedup compared to no OC		
	ISO 14443A	ISO 15693
+1 Mhz	7	7
+2 Mhz	13	13
+3 Mhz	20	21

Table 2: Tag response speedups for various overclocks.

The results show similar speedups for ISO 14443A and ISO 15693. For every Mhz increase in clock speed, we see a speedup of between 6% and 8%, though this pattern will of course not continue forever.

For ISO 15693, we also tested whether overclocking had the expected impact on the response acceptance rate. Specifically, we compared response acceptance rates for various overclocks to the acceptance rate with no overclock. We disabled the simulator's extra leeway time and used a 10 μs relay delay in our tests. As mentioned in section 5.1.1, responses in this scenario were almost always rejected when no overclock was used. By contrast, a 2 Mhz overclock increased the response acceptance rate to 55%.

It is safe to say that tag overclocking is likely to lower relay delay significantly.

6 Discussion

Our results suggest that the RFID Guardian is certainly feasible as a relay attack platform. Attacks against ISO 14443A seem very much achievable. Although ISO 15693 appears to be more difficult to attack, we do not rule out the possibility that it can be relayed succesfully as well with the help of tag overclocking. It is not very difficult to implement a relay attack on the RFID Guardian in the manner we described in the previous sections, but the implications are potentially severe. We illustrate this with a number of examples of potential relay attacks against real–life RFID systems.

All e–passports in the European Union adhere to the ICAO 9303 standard for electronic passports [ICA06], and passports in the United States are likely to follow. Since ICAO 9303 requires the use of ISO 14443 type A or B, essentially all EU passports are vulnerable to relay attacks. Using ISO 14443 type B instead of type A does not help, as type B lets the tag determine the Frame Waiting Time just like type A does. Switching to an entirely different protocol has become extremely difficult and costly, because many countries have already implemented e–passports and set up the corresponding architecture at their airports.

Many airports, including Amsterdam Schiphol Airport, are now experimenting with self–service check–in [Sch08]. Self–service check–in machines scan e–passports automatically, without a human customs officer checking them. This means that an attacker could relay communications between the self–service check–in machine and the passport of another passenger at the airport. This would allow the attacker to travel in the other passenger's name.

Several hospitals are experimenting with the use of RFID to identify patients [FM08]. A malicious attacker could remotely disable a victim patient's legitimate RFID tag, and then setup a relay channel between another patient's tag and a hidden proxy tag near the victim patient. This could cause wrong medication or treatment to be administered to the victim patient, potentially resulting in injury or even death.

RFID–based payment systems are also being developed. Some of these systems even allow small transactions without requiring a PIN–code. For example, MasterCard's "PayPass" system will allow transactions below a regionally specified limit without requiring any user confirmation [Pay08]. This limit is typically in the order of $50. PayPass uses the ISO 14443 protocol, and is thus vulnerable to relay attacks. The PayPass system is already in use at various locations, and is currently being introduced in the Netherlands as well by the ABN-AMRO bank.

Such payment systems essentially allow remote pickpocketing. An attacker could simply stand in a busy mall with a proxy reader, placing the proxy tag by his own payment machine. It is then possible to perform transactions with the payment cards of unsuspecting people visiting the mall. Stealing a small amount of money from every card is unlikely to be noticed quickly. Many such small amounts can soon add up to a lot of money, making this a low–risk high–profit attack.

Of course, an attack against a building's access system as proposed in section 1 is also entirely feasible. With increasingly many company, government and military installations using such systems for building access control, the door is literally wide open to both corporate and military espionage.

7 Conclusion

Relay attacks are quite easy to execute. The consequences, however, can be extremely serious. Additionally, relay attacks can be highly profitable and induce low risk for an attacker, making them very attractive as RFID becomes more and more embedded in society.

Today's RFID standards do not pay much at-

tention to the risk of relay attacks. Although it is possible to defend a system against such attacks, this is difficult, and it involves methods which are not employed in any real RFID systems today.

The RFID Guardian seems entirely feasible as a relay attack platform, and implementing a relay attack with it is quite easy. As the RFID Guardian becomes commercially available, it can be used for both good and evil. People can use it to protect their privacy and RFID tags, yet it can also serve as an effective attack platform against these same people. However, this does not mean the RFID Guardian should not be made commercially available: without such a device, the public would have no means of defense at all.

One thing is indisputable: RFID is becoming ever more omnipresent, and we can not afford to ignore the risk of relay attacks any longer.

References

[AI08] Syed Ahson and Mohammad Ilyas. *RFID Handbook: Applications, Technology, Security and Privacy.* CRC Press, 2008.

[Blu09] Bluetooth specification version 3.0, 2009.

[DM07] Saar Drimer and Steven Murdoch. Keep your enemies close: Distance bounding against smartcard relay attacks. In *Proceedings of the 16th USENIX Security Symposium (SS '07)*, pages 1–16, Berkeley, CA, USA, 2007. USENIX Association.

[FM08] Jill Fisher and Torin Monahan. Tracking the social dimensions of RFID systems in hospitals. *International Journal of Medical Informatics*, 77(3):176–183, 2008.

[Han05] Gerhard Hancke. A practical relay attack on ISO 14443 proximity cards. Manuscript, February 2005.

[HK08] Gerhard Hancke and Markus Kuhn. Attacks on time-of-flight distance bounding channels. In *Proceedings of the first ACM Conference on Wireless Network Security (WiSec '08)*, pages 194–202, Alexandria, Virginia, USA, March-April 2008. ACM, ACM Press.

[HMM09] Gerhard Hancke, Keith Mayes, and Konstantinos Markantonakis. Confidence in smart token proximity: Relay attacks revisited. *Computers & Security*, 28(7):615–627, 2009.

[HR07] Martin Hlaváč and Tomáš Rosa. A note on the relay attacks on e-passports: The case of Czech e-passports. Cryptology ePrint Archive, Report 2007/244, 2007.

[ICA06] Machine readable travel documents (International Civil Aviation Organization (ICAO) doc 9303), 2006.

[ISO00] ISO/IEC 15693, vicinity cards, first edition, 2000.

[ISO01] ISO/IEC 14443, proximity cards, first edition, 2001.

[KW05] Ziv Kfir and Avishai Wool. Picking virtual pockets using relay attacks on contactless smartcard systems. In *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM '05)*, pages 47–58, Washington, DC, USA, 2005. IEEE Computer Society.

[OW10] Yossef Oren and Avishai Wool. RFID-Based electronic voting: What could possibly go wrong? In *International IEEE Conference on RFID*, pages 118–125, Orlando, USA, 4 2010.

[Pay08] PayPass mag stripe national merchant implementation requirements version 1.0, 2008.

[RCT05] Melanie Rieback, Bruno Crispo, and Andrew Tanenbaum. RFID guardian: A battery-powered mobile device for RFID privacy management. In *Proceedings of the 10th Australasian Conference on Information Security and Privacy (ACISP '05)*, volume 3574 of *Lec-*

ture Notes in Computer Science, pages 184–194. Springer-Verlag, July 2005.

[RGC+06] Melanie Rieback, Georgi Gaydadjiev, Bruno Crispo, Rutger Hofman, and Andrew Tanenbaum. A platform for RFID security and privacy administration. In *Proceedings of the USENIX/SAGE Large Installation System Administration conference (LISA '06)*, pages 89–102, Washington DC, USA, December 2006.

[Sch08] Schiphol airport self-service check-in trial (www.airport-int.com), 2008.

[SVN] RFID guardian SVN repository (http://www.rfidguardian.org/websvn).

[USB00] Universal Serial Bus specification version 2.0, 2000.