Chapter 10
Security in RFID Networks and Communications

Chiu C. Tan[1] and Jie Wu[2]

Abstract

Radio frequency identification (RFID) networks are an emerging type of network that is posed to play an important role in the Internet-of-Things (IoT). One of the most critical issues facing RFID networks is that of security. Unlike conventional networks, RFID networks are characterized by the use of computationally weak RFID tags. These tags come with even more stringent resource constraints than the sensors used in sensor networks. In this chapter, we study the security aspects of RFID networks and communications. We begin by introducing the main security threats, followed by a discussion of various security mechanisms used to protect RFID networks. We conclude by studying the security mechanism of an actual large scale RFID deployment.

10.1 Introduction

Radio frequency identification (RFID) technology consists of small inexpensive computational devices with wireless communication capabilities. Currently, the main application of RFID technology is in inventory control and supply chain management fields. In these areas, RFID tags are used to tag and track physical goods. Within this context, RFID can be considered a replacement for barcodes.

RFID technology is superior to barcodes in two aspects. Firstly, RFID tags can store more information than barcodes. Unlike a barcode, the RFID tag, being a computational device, can be designed to process rather than

[1] Temple University, Philadelphia, PA, USA. E-mail: cctan@temple.edu.
[2] Temple University, Philadelphia, PA, USA. E-mail: jjewu@temple.edu.

L. Chen et al., *Wireless Network Security*
© Higher Education Press, Beijing and Springer-Verlag Berlin Heidelberg 2013

just store data. Secondly, barcodes communicate using an optical channel, which require the careful positioning of the reading device with no obstacles in-between. RFID uses a wireless channel for communication, and can be read without line-of-sight, increasing the read efficiency.

The pervasiveness of RFID technology in our everyday lives has led to concerns over whether these RFID tags pose any security risk. For example, consider an RFID tag affixed to clothing, this type of tag contains information such as the brand and model of the clothing. This type of information is used for inventory purposes. A thief armed with an RFID reader can, however, use the same information to select wealthy targets, which are more likely to wear more expensive clothes, to pickpocket.

The future applications of RFID make the security of RFID networks and communications even more important than before. The ubiquity of RFID technology has made it an important component in the Internet-of-Things (IoT), a future generation Internet that seeks to mesh the physical world together with the cyber world[1]. RFID is used within the IoT as a means of identifying physical objects. For example, by attaching an RFID tag to medication bottles, we can design an RFID network to monitor whether patients have taken their medications. RFID readers can be used to determine when medication bottles have been removed from the medicine cabinet, this information can be combined with additional information, such as weight sensors that record the weight of medicine bottle, to infer whether a patient has taken his medication. Such applications, while undoubtedly useful, opens the door to allow malicious entities to launch attacks like determining what types of medication a person is taking.

Given the stakes, it is unsurprising that RFID security has attracted the attention of researchers. In recent years, there have been numerous RFID security protocols proposed, and new RFID vulnerabilities discovered. The difficulty in securing RFID lies in the resource constraints of the RFID tags, which makes it impossible to adopt existing security solutions from other fields such as mobile computing or wireless networking, onto RFID networks.

This chapter studies the security of RFID networks. Firstly, we discuss some background on RFID networks, followed by an introduction to main RFID threats. We then review and analyze some basic RFID security protocols, followed by a discussion on more advance attacks and defense. Finally, we discuss the security of industry standard RFID protocols.

10.2 RFID Network Primer

An RFID network consists of three basic components: RFID tags, RFID readers, and backend servers. In an RFID network, each RFID tag contains small amounts of information which are affixed to physical objects. RFID readers read the information from these tags as the physical object moves around a given area. The information is then transmitted from the readers to backend servers for processing to service higher level applications. Fig. 10.1 shows the interactions between the three components.

Fig. 10.1 shows that all interactions are reader driven. The RFID tag never initiates any communications. The RFID reader can be configured like a Wi-Fi access point (AP) beaconing to periodically broadcast a query to read tags in the vicinity, or the query can be manually triggered. The communication channel between the RFID reader and the backend server can be either wired or wireless, and is assumed to be secure. We also assume that some access control policy is in place to regulate reader access to the backend server. The channel between the reader and the tag is assumed to be insecure. The majority of RFID security research is focused on securing this wireless channel.

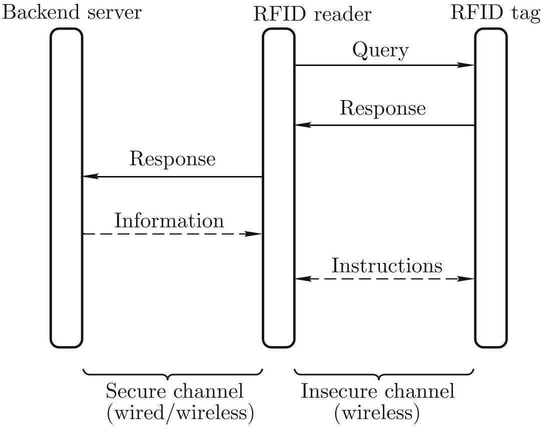

Fig. 10.1 Basic interaction between the components. Dashed line indicates optional operations. There can be multiple interactions between the reader and tag in the "Instructions" command, as denoted using double headed arrows.

The two optional operations, shown the Fig. 10.1, are generally used when the reader needs to write any information onto the RFID tag. To protect the integrity of the RFID tag data, writing to the tag's memory typically requires some sort of password which is stored in the backend servers.

An example of an RFID network is an RFID-enabled hospital. Patients are given a unique RFID tag to wear. The tag contains the patient's unique ID. RFID readers installed throughout the hospital can track the movement of patients through the reading of the tag IDs. In addition, medical treatments (e.g., blood bags, pills, etc.) are also embedded in RFID tags. The backend servers will associate a patient's RFID tag ID with the appropriate treatments, and a nurse will scan all the tags before administrating treatments.

10.2.1 RFID reader characteristics

The RFID network may consist of both mobile and static RFID readers. The mobile reader combines a processing unit and antenna together, and resembles a smartphone type device. The processing unit is used to communicate with the backend server and issue commands to the antenna. The antenna is used to broadcast and receive messages. A user will aim the reader at a set of RFID tags to query them. Information from the mobile reader can be transmitted to the backend servers wirelessly. A static RFID reader has the antenna permanently positioned at a specific location (i.e. entrance to a specific hallway). Multiple antennas may share a single processing unit, and these antennas are connected to the processing unit via wired channel.

We usually do not make a distinction between the antenna and the processing unit in RFID security literature. They are both simply referred as "RFID reader". However, this distinction can be important for performance reasons where angles of the antennas matter.

The purpose of the RFID reader is to communicate with RFID tags, and send the information back to the backend servers. Besides, the reader is responsible for regulating tag responses. One of the limitations of the RFID tag is that the tag cannot perform carrier sensing. Instead, the RFID reader acts as a coordinator to regulate tag communications. Most RFID security protocols however ignore this function and simply assume that there is only a single reader querying a single tag.

10.2.2 RFID tag characteristics

Each RFID tag contains a unique identifier (id). Once a tag is affixed to a physical object, the id becomes a representation of that object.

1. *Types of RFID tags*

There are three general types of RFID tags, active, semi-active, and passive RFID tags.

- *Active RFID tags.* This type of RFID tag contains an internal battery which is used to let the tag perform more complex operations, such as monitor temperature, as well as boost the communication with an RFID reader. The communication range of an active tag can be over 100 meters. An active tag is the most powerful type of RFID tag, and is also the most expensive.
- *Semi-active RFID tags.* This type of tag also contains an internal battery, but unlike an active tag, the battery is only used for the tag's internal operations, and not for communication. A semi-active RFID tag relies on RFID reader to supply the necessary power for communication. Note that semi-active tags are sometimes known as semi-passive tags.

- *Passive RFID tags.* This type of RFID tag have the lowest cost (pennies per tag), and unsurprisingly, are the most prevalent type of RFID tags. A passive tag has no internal batteries, and relies on the RFID reader to supply the power needed to perform all tag operations and communication. In the rest of this chapter, our focus is on this type of tags.

2. *Communication range*

The conventional range of the tag can range from several centimeters, for RFID tags operating in the 13.56 Hz, to over a dozen meters for RFID tags operating in the 902-928 MHz. Due to the physical characteristics of the reader and tag, the signal being passed from the reader to the tag is stronger than that from the tag to the reader. This means that for certain operations like eavesdropping, it will be easier to hear the RFID reader's commands than it is for the tag's response.

In terms of security, however, we cannot rely on the conventional communication range. Determining the RFID tag communication range for security purposes is difficult for two reasons. Firstly, RFID tag responses are sensitive to environmental conditions. Reading an RFID tag on credit card in a purse placed in a handbag is very different from reading a tag placed on a store shelf. Secondly, when launching an attack, the adversary can use non-standard equipment that is more powerful than regulation equipment. There have been experiments on querying RFID tags in "realistic" environments (tag placed in a person's wallet), but such experiments are limited by the use of conventional equipment[2].

3. *Computational ability*

Despite having no battery power, passive RFID tags do exhibit a wide range of capabilities. RFID tags contain limited amounts of persistent storage capacity, and the storage on a tag can be read-only, write-once, or multiple writes. The difference between a read-only and a write-once tag is that for a read-only tag, the initial information is usually stored when the tag is manufactured and not transmitted by a RFID reader. The distinction between tags that support multiple writes and those that do not is important for RFID security, since some protocols require authorized readers to change the stored data after every successful read. Current commercial RFID tags can perform functions such as matching bit strings, exclusive-ORs, generate random numbers, cryptographic hash functions, and symmetric key operations. Within RFID security research, there is work on designing security solutions that do not use these functions. For instance, there are protocols that use only hash functions, or do not use random number generators. The reason is that engineering more functions will increase the cost of the tag, and using weaker tags are cheaper. Table 10.1 lists the capabilities of a sample RFID tag.

Table 10.1 Sample RFID tag security capability

Type of tag	Security capabilities
Low end	32-bit access and kill password. The kill password is used to render the RFID tag non-responsive to further RFID reader queries. 64-bit fixed ID value. ID assigned at time of manufacture, and cannot be changed. Used for certain counterfeit tracking operations.
High end	64-bit mutual authentication protocol (proprietary). Stream encryption capabilities (proprietary). Support multiple passwords for fine grain access control. Different memory locations can require different passwords to access.

1 Based on EPC Class 1-Gen 2 standards and Alien Technology ALN-9640 tags.
2 Based on Atmel ATA6286 Crypto RF tag.

10.3 Security Requirements

The challenges in security RFID networks lie in securing the operations involving RFID tags. This is because the severe resource limitations of tags make it difficult to implement conventional security mechanisms. RFID readers and backend servers on the other hand, can be secured using existing security techniques. In this section, we begin by examining the key RFID security requirements, followed by more specific requirements for certain RFID applications.

There are three key RFID security requirements: prevent unauthorized access, prevent illicit tracking, and prevent or detect skimming. These form the basic requirements for most RFID applications.

- *Prevent unauthorized access.* There are two ways which unauthorized access can occur. The first is when an unauthorized RFID reader queries and obtains usable information such as the tag ID from the RFID tag. RFID tag design requires the tag to respond to any query. Any reader can query the tag and get a response. Preventing unauthorized access refers to allowing only authorized readers to obtain usable information. The second way which unauthorized access can occur is via eavesdropping. An adversary obtains usable information by observing the over-the-air communications between a legitimate reader and a tag.
- *Prevent illicit tracking.* This requirement addresses one of the main privacy concerns over the use of RFID technology. Illicit tracking exploits the fact that RFID tags always respond to reader's query. An adversary that queries and obtains the same tag response at multiple locations can infer that the same tag has visited those locations. Since RFID tags are affixed to physical objects, for instance clothing, this implies that the same person has visited those locations. Note that satisfying the first requirement does not automatically satisfy this requirement. A tag that returns a constant, encrypted response will prevent unauthorized access, since the adversary cannot determine the tag contents. However, the constant ciphertext can be used to perform illicit tracking.
- *Prevent or detect skimming.* Skimming is an attack whereby the adversary

observes the interactions between a legitimate RFID reader and a tag, and tries to create a fake RFID tag that mimics a real one. The adversary succeeds when his fake tag can pass off as a real tag. Skimming is a concern when RFID is used to authenticate documents such as driver licenses or passports. For instance, an adversary that tries to create a fake drivers license may attempt to observe the interactions of an RFID tag embedded in a legitimate drivers license to create his fake RFID tag. Generally, the adversary performing a skimming attack does not have physical access to the RFID tag.

In addition to the key requirements listed above, there are more specific security requirements that are important for certain applications. Applications that transfer ownership of the tag, either temporarily or permanently will require that the previous owner of the RFID tag can no longer access the data stored in the tag. This requirement is known as secure ownership transfer. A related requirement is forward-security. This requirement means that an adversary that learns of an RFID tag secret, for instance, by physically compromising the tag, cannot determine previously encrypted information from that tag. Secure RFID search is used when a user wishes to locate a particular tag from a large collection of RFID tags. The requirement of a privacy-preserving RFID search is to ensure that searching does not leak information about the RFID tag.

There are two advance requirements that cannot be easily solved using the solutions address on the basic requirements.

- *Defend against Mafia fraud.* This is a relay-type attack where the adversary deploys a fraudulent reader and tag. The fraudulent reader will query the real RFID tag, and then relay the information to the fraudulent tag to replay to a legitimate reader. Defense against this type of attack is needed for applications that use RFID tags for access control purposes (e.g. opening a car door), or for payment applications like credit cards.
- *Grouping proofs.* A grouping proof requires the RFID reader to prove that a set of RFID tags were read at the same time and location. For instance, a patient may be required to take three types of medications at the same time. The nurse with a mobile RFID reader can generate a grouping proof that captures all three RFID tags (affixed to the medication containers) were present at the same time to prove that the patient was correctly medicated.

10.4 Hardware Based Solutions

A straightforward solution to provide security is to physically disable the RFID tags. The idea of a “clipped tag” was proposed where the RFID tag was designed to allow the user to separate the RFID chip (contain the tag data) from the tag antenna (used to power the chip)[3]. This way, no RFID readers can query the tag, and thus making it secure. Later work improved upon this idea by allow the clipped tag to continue to be read by an RFID reader, but at a much shorter distance[4]. This approach resolves the key RFID security requirements by forcing adversary to be physically very close to the RFID tag to read any data, which makes such attacks easily detectable.

An important argument against disabling the RFID tag is that the process is irreversible. Instead, an alternative is to design a special device to disrupt the RFID operations, which a user can carry with them. This idea was first proposed by Juels et al. in the form of a blocker tag, a special RFID tag which can be programmed to block certain tag IDs that the user considers sensitive[5]. The blocker tag is also a passive RFID tag. Feldhofer et al. proposed a watchdog type device to alert users when an RFID reader is querying their tags[6]. Later work by reference [7] developed a more powerful battery operated device, the RFID Guardian, that intercepts the RFID reader's signal and only allow signals from authorized readers to reach the tag. Since the adversary never gets any response from the RFID tag, the guardian provides the needed security requirements.

Hardware based solutions, while being an important component in RFID security research, are less popular than protocol based solutions. There are several possible reasons for this. Firstly, hardware type solutions tend to be more expensive due to the use of external devices. Secondly, such solutions can potentially disrupt operations of other RFID tags belonging to other users, which make it more difficult to gain acceptance. Finally, when RFID tags were initially deployed, there were concerns that tag manufacturers may be unwilling to engineer security protections into the tag since this will increase their manufacturing cost. Hardware based solutions are practical in that context since they do not rely on the tag manufacturers. In recent years however, public awareness over security RFID appear to have led to the deployment of more secure RFID tags, making hardware based solutions less attractive.

10.5 Basic Protocol Based Solutions

Protocol based RFID solutions rely on the RFID tags performing certain operations to provide the key security requirements to prevent unauthorized access, tracking, and skimming.

10.5.1 Different RFID Protocols

There are too many RFID protocols in the literature to be included in this chapter. Instead, we attempt to categorize them based on the focuses of these protocols, and highlight just a few works in each category. We have elected to avoid discussing RFID protocols designed from specialized applications such as banknotes[8] or supply chains[9]. A good resource for the latest updates can be found in reference [10].

1. *Improving backend performance*

One approach lies in improving the performance of the backend server. From Fig. 10.2, we see that the backend server needs to try all $(s : id)$ pairs to determine the correct secret s to use in order to obtain the tag id. The reason that the RFID tag does not inform the backend server which secret s to use is to defend against illicit tracking. As a result, the RFID tag has to output a different random value each time it is queried. A more detailed analysis of protocols designed to alleviate the bottleneck can be found in reference [11].

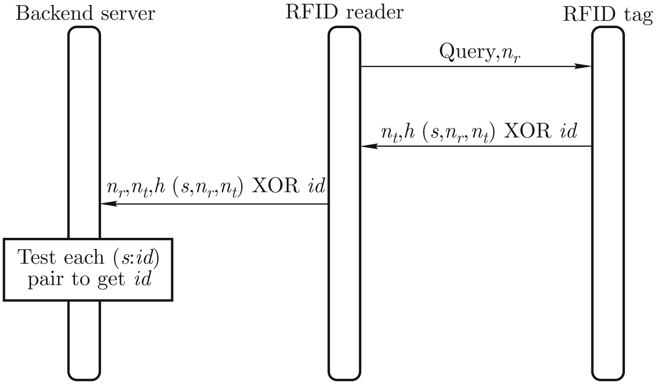
Backend server
RFID reader
RFID tag
Query,n_r
n_t,h (s,n_r,n_t) XOR id
n_r,n_t,h (s,n_r,n_t) XOR id
Test each (s:id) pair to get id

Fig. 10.2 Basic protocol that defends against key RFID security requirements. Modified from Tan et al.[22]. Random numbers from the reader and the tag are denoted as n_r and n_t respectively. The variables s and id denote the RFID tag's secret and id. Each tag has a unique secret and id that is assigned by the backend server. A conventional hash function is denoted as $h()$.

One example of such protocols is a time-based solution proposed by reference [12]. The intuition is to let the backend server maintain a lookup table associated with the tag secret that is hashed with a timestamp, $(h(s,t) : id)$. The backend server can pre-compute this table each time t. Each time the reader queries the tag, the reader will send the timestamp t, and the tag will respond with $h(s,t)$. This way, the backend server can obtain the corresponding id immediately using the lookup table. Later work by reference [13] and reference [14] improves on this approach.

2. *Using lightweight primitives*

In Fig. 10.2, the RFID tag uses a hash function $h()$ to protect its response. Given the hardware limitations of the RFID tag, an area of RFID research attempts to design solutions that do not use hash functions or symmetric keys to provide security. One popular approach is generally known as HB family of protocols which is after the authors[15]. The HB family of protocols uses scalar products and exclusive-ORs to design their protocols. Work by reference [16] first proposed HB protocols that defend against different RFID attacks. A general survey of the HB family can be found in reference [17].

3. *Generating random numbers*

RFID protocols make extensive use of random numbers. A weak source of random numbers will allow the adversary to launch tracking. The use of random numbers to defend against tracking depends on the quality of the random number generated by the weak RFID tag. Work by Holcom et al.[18], J. Melia et al.[19], and Peris et al.[20] explores this problem in further detail.

10.5.2 A Detailed Look at a Simple RFID Protocol

Here we introduce a protocol modified from Tan et al.[21] to illustrate how a protocol based solution provides key RFID security requirements. Table 10.2 lists the notation used in this chapter. Fig. 10.2 illustrates the protocol.

Table 10.2 Notations used

n_t	Random number generated by RFID tag
n_r	Random number generated by RFID reader
s	RFID tag secret
id	RFID tag id
$h()$	Cryptographic hash function
t	Timestamp

From the protocol shown in Fig. 10.2, we see that when the reader queries the tag, the reader will first transmit a random number, n_r, to the tag. The RFID tag will respond by first generating its own random number, n_t, and then compute a response to protect its tag *id* using $h(s, n_r, n_t)$ XOR *id*. The reader then re-directs the tag's response, together with the random number it chose, to the backend server.

The role of the backend server is to determine the *id* of the RFID tag. Since the backend server is responsible for all the RFID tags, it maintains a list of all tag secret to tag *id* pairs $(s : id)$. Upon receive the message for the RFID reader, the backend server will know the two random numbers n_t and n_r chosen by the tag and reader respectively. From the list of $(s : id)$ pairs, the backend server will hash the secret s to generate $h(s, n_r, n_t)$ and XOR that against the response by the RFID reader, i.e. $h(s, n_r, n_t)$ XOR $h(s, n_r, n_t)$ XOR *id*. If the result matches the *id* in the list, the backend server will have determined the tag *id*. Otherwise, the backend server will continue to the next pair.

10.5.3 Security analysis

Here we will analyze how the basic protocol in Fig. 10.2 meets the key RFID security requirements.

The first requirement is to prevent unauthorized access to the RFID tag information. We first consider an unauthorized reader querying the tag. The adversary will issue its own random number n_r, and receive n_t, $h(s, n_r, n_t)$ XOR *id* from the tag. Since the backend server will not respond to the adversary, the adversary now has to determine *id* without any help from the backend server. The adversary succeeds if he is able to determine *id* from n_t, $h(s, n_r, n_t)$ XOR *id*. In order to get back *id*, we need to XOR with $h(s, n_r, n_t)$ using the correct s value, but the adversary only knows n_r and n_t, and not s. Thus, the adversary is unable to obtain *id*. Since the protocol uses a conventional hash function such as SHA, the adversary cannot obtain s from $h(s, n_r, n_t)$. The adversary can attempt to guess the value of s, but this can be defended against by using large enough values of s.

Another method of unauthorized access is for the adversary to be eavesdropping when a legitimate reader is querying a tag. Since the wireless channel between the reader and tag is assumed to be insecure, the adversary is able to learn n_r, n_t, and $h(s, n_r, n_t)$ XOR *id*. These pieces of information are similar to that obtained when the adversary queries the tag directly, which yields no useful information to the adversary.

The second requirement is to prevent illicit tracking. In this attack, the adversary needs to determine whether two tag responses belong to the same RFID tag. From Fig. 10.2, we see that the RFID tag has two pieces of information that remains constant, the tag *id* and tag secret s. However, each time the tag replies to a query, the tag will select a different random number, n_t, and thus, the resulting $h(s, n_r, n_t)$ XOR *id* will always be different for every response. This prevents any illicit tracking, since the adversary is unable to determine whether two responses are from the same RFID tag or not. This defense remains valid even if the adversary can select its own n_r value.

The third key requirement is to prevent or detect skimming. The adversary launching a skimming attack will observing the responses of a real RFID tag in attempt to create a fake tag that can pass off as a real tag. In the basic protocol, the adversary is able to observe the return value of n_t, $h(s, n_r, n_t)$ XOR *id*, However, it is unable to learn s or *id* based on the response. The adversary thus can only store $h(s, n_r, n_t)$ XOR *id* directly into a fake RFID tag. This skimming attack will be detected when a legitimate reader queries the RFID tag. The legitimate reader will issue its own random number, which we denote as n'_r to distinguish from the earlier n_r observed by the adversary. Since the fake tag does not know s or *id*, the fake tag can only return $h(s, n_r, n_t)$ XOR *id*, and not the correct $h(s, n'_r, n_t)$ XOR *id*. Since the backend server will attempt to test using n'_r and not n_r, this leads the backend server unable to find a correct (s, id) pair. Thus the skimming attack is detected.

10.6 Advance Protocol Based Solutions

Beyond the key RFID security requirements, there are some other RFID security requirements. This section discusses some protocols that address these requirements. Note that the protocols presented here may not necessary meet all the key security requirements because these advance protocols are generally designed to address specific issues or applications.

10.6.1 Defending against Mafia fraud

The mafia fraud has emerged as a challenging problem for RFID applications. This type of attack cannot be defended by the protocols mentioned earlier because a legitimate RFID reader is accessing data from a legitimate RFID tag. In other words, this type of attack can still work even if both the reader and the tag authenticate each other. This is illustrated using the basic protocol shown in Fig. 10.2.

Consider an application which uses RFID tag to open a door. The RFID reader will first read the tag and then transmit the information to the backend server. Once the backend server verifies the tag is legitimate, the door will open. To launch a mafia fraud attack, the adversary will first be in close proximity with a person holding a legitimate RFID tag. We refer to this person as the target. The adversary's accomplice will be standing near to the door. When the legitimate RFID reader issues a query, the adversary's accomplice will relay this message to the adversary, who will in turn issue it to the target's RFID tag. The target's RFID tag will respond to the adversary, who will then relay this back to his accomplice to transmit to the RFID reader. Since the RFID reader obtains the response from a legitimate RFID tag, the door will open and the adversary can gain access. Therefore the choice of protocol does not defend against this type of attack.

The intuition behind the defending against a mafia fraud is to accept an RFID tag's response if it is both valid and timely. Since the wireless transmission speed, the RFID tag computational time, and distance between the reader and tag are known, the RFID reader can estimate the amount of time needed to receive a response. If the arrival of the RFID tag response is late, the reader can deduce the distance travelled is longer than what is allowed, and thus reject the tag answer.

One of the main solutions against the mafia fraud is from Hancke et al.[23] and is shown in Fig. 10.3. We assume the system will define a maximum distance d, over which the reader is not suppose to authenticate a tag. In the protocol, we see that both reader and tag exchange random numbers. Assuming that the reader knows the tag secret s, both entities can compute $h(s, n_r, n_t)$. The tag will split this result into two queues, X and Y. At the same time, reader than generates a k bit challenge, $C_1, \cdots, C_k$, where k is a system defined parameter. The idea is that the reader will challenge the tag by sending over a bit C_i. If the C_i is 0, the tag will set A_i to the bit from queue X, and vice versa. The reader will keep the time it takes from sending C_i and receiving the A_i. A legitimate RFID tag that is within the approved distance will respond within the allocated time limit. A legitimate RFID tag that is further away will take a longer time to respond, due to the longer distance travelled, and thus is detected. More recent work on this topic can be found in reference [24] and reference [25]. An interesting idea of doing distance bounding for a group of RFID tags instead of just two tags has been proposed by Capkun et al.[26].

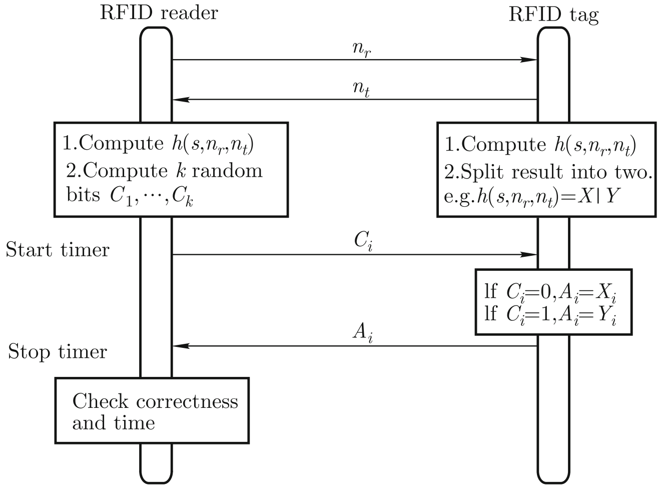

Fig. 10.3 Distance bounding protocol from Hancke et al. We assume that the reader already knows the tag secret s. We retain the notation from Table 10.2. X and Y are the bitstring resulting from dividing $h(s, n_r, n_t)$ into two. The protocol will repeat itself until C_k is transmitted from the reader to the tag. The variable k is a system defined parameter.

10.6.2 Grouping proofs

Grouping proofs are required for a reader to prove to the backend server that a set of RFID tags are physically close to each other. This type of proof typically requires a more advanced RFID tag that is able to maintain an atomic counter and a countdown timer. Each time an RFID tag is queried, the RFID tag will increment its counter after its timer expires. This is an atomic operation that cannot be disrupted. The intuition is for the reader to query each RFID tag one after the other to collect the responses to generate a proof before the timer expires.

Fig. 10.4 illustrates a grouping proof from Bolotnyy et al.[27]. The proof is to demonstrate that RFID Tag 1, Tag 2, and Tag 3 are present at the same time. To generate the proof, the reader will first query the first tag, Tag 1, and receive a_1 where $a_1 = \{id_1, c_1, h(s_1, c_1)\}$. At this point, the timer for Tag 1 has started. The reader will continue to send a_1 to Tag 2 and receive a_2 back (Step 2). The value of a_2 is $\{id_2, c_2, h(s_2, c_2, a_1)\}$. The reader will send a_2 to Tag 3 and get back a_3 (Step 3), which is $\{id_3, c_3, h(s_3, c_3, a_2)\}$. At this time, the reader has collected responses from all the tags, and will send a_3 back to the first tag, Tag 1. This has to be done before Tag 1's timer expires. If the reader is successful, the reader will obtain the message m, where $m = h(a_1, a_3, s_1)$. RFID Tag 1 will not respond if the timer expires. The reader will then submit the proof p to the backend server for verification, where $p = \{id_1, id_2, id_3, c_1, c_2, c_3, m\}$. Since the backend server knows the secrets for each of the ids, the backend server can determine whether c_1, c_2, c_3, and m are valid.

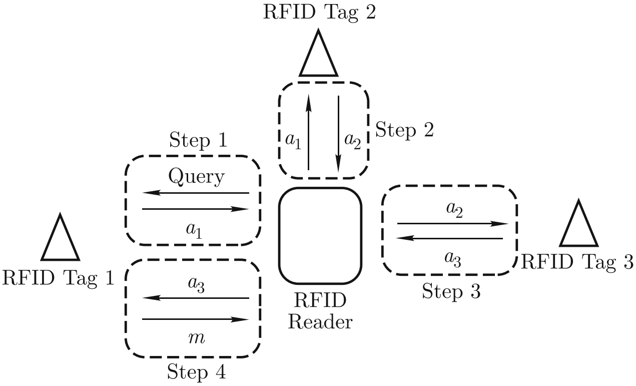

Fig. 10.4 Grouping proof for 3 RFID tags. After Step 4, the reader generates the proof, which is then transmitted to the backend server for verification. Steps 1 to 4 have to be completed before the RFID tag timer expires.

We can see that if the RFID reader does not complete the proof in time, the reader will be unable to return the correct m value to the backend server because computing m requires s_1, which is only known to the Tag 1 and the backend server. The reader also cannot reuse old values such as a_1, a_2, or a_3, since the counter value for each tag will increment each time, creating an incorrect m value. Grouping proofs are also known as "yoking-proofs", which was first proposed by Juels[28], which as limited to 2 tags. More recent work by Burmester et al.[29] and Tan et al.[21] improves on this concept.

10.7 Commercial RFID Security

In this section, we turn our focus to commercial RFID security solutions. Details regarding commercial RFID systems are often difficult to come by, since companies are reluctant to release information publically. Despite this, researchers have been successful in reverse engineering some RFID products. Recent work by Garcia, et al.[30], Kasper et al.[31], and Nohl, et al.[32] have demonstrated vulnerabilities in some commercial RFID systems.

Here, we consider the security mechanisms for RFID enabled passport (ePassport). Since passports have to be interoperable among various airports globally, documentation on the security mechanisms is available.

10.7.1 Background on RFID-enabled Passports

The standards for RFID-enabled passports are maintained by the International Civil Aviation Organization (ICAO), which maintain, among other things, the protocols needed to access the RFID tag embedded within passports. Since our focus is on RFID systems, we limit our discussion to the common interaction between the RFID reader and the tag. Details such as maintaining public key infrastructure (PKI) and RFID reader revocation are omitted. Interested readers can obtain more information from International Civil Aviation Organization documentation[33].

The RFID tag within the passport contains data relating to the passport holder, for instance, the height or photograph of the passport holder. Since the RFID tag has limited storage capacity, a hashed result of such information is stored in the RFID tag. The basic steps for verifying a passport is given in Fig. 10.5. These steps are performed, for instance, at the immigration counter at an airport.

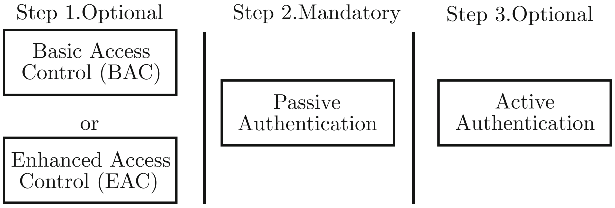

Fig. 10.5 Basic steps when RFID reader queries the RFID tag in the passport. At the end of Step 3, the RFID reader will determine whether the tag is valid or not.

The first step is supposed to regulate access to the data contained within the RFID tag. There are two types of access control, the basic access control (BAC) and the extended access control (EAC). According to Chothia et al.[34], most passports already implement BAC. We will focus our discussion on BAC in the next subsection.

The second step in the verification process is mandatory. The purpose of passive authentication is to verify that the data contained within the RFID tag is valid. When the passport is first issued, information about the passport holder is hashed and signed with a secret key that is associated with the country who issued the passport. In the passive authentication step, the hashed data and signatures are verified using the public key associated with the country.

The last step, active authentication, is needed because Step 2 only verifies that the data contained in the RFID tag is genuine. It does not indicate that the passport itself is legitimate. The reason is that an adversary could skim the data off the real passport, and stored it into the RFID tag of a fake passport. In Step 3, the RFID tag itself is authenticated. Performing step 3 requires the RFID tag to perform public key operations. In active authentication, the RFID reader will send a random number over to the RFID tag, which will then digitally sign the number and return the signature to the reader. Active authentication is also optional process.

A passport that only implements the mandatory passive authentication does not satisfy the three key RFID security requirements discussed earlier. From International Civil Aviation Organization document[33], the motivation for implementing BAC is to prevent skimming and eavesdropping. Even though preventing illicit tracking is not a stated goal of BAC, we will see in the later security analysis, BAC also protects against tracking.

10.7.2 Basic Access Control Protocol

An RFID tag that runs BAC has to be able to perform symmetric key operations. The tag will store two symmetric keys permanently, K_{enc} and K_{mac}. These two keys are computed when the passport is first issued to the passport holder. The goal of BAC is to allow the RFID reader and the RFID tag to eventually derive a session key KS_{enc} and KS_{mac} to encrypt future transactions.

In the basic RFID protocol introduced earlier, a challenge in RFID protocols is to efficiently determine which secret is associated with a particular tag. A similar problem is encountered here, where the RFID reader has to determine which K_{enc} and K_{mac} belongs to the passport. The RFID enabled passport overcomes this problem by computing K_{enc} and K_{mac} using a function of
<Passport number, Passport holder's date of birth, and Passport's Expiry date>.

All passports must contain these three pieces of information. The reasoning is that these information can be easily obtained when a passport holder gives his passport to immigration personal for verification, upon which the RFID reader can obtain K_{enc} and K_{mac}. Assuming that the RFID reader now posses K_{enc} and K_{mac}, Fig. 10.6 shows the rest of the BAC protocol.

After the RFID reader issues a query, the tag will respond with a random number n_t. The reader will execute Step A, which consists of the following

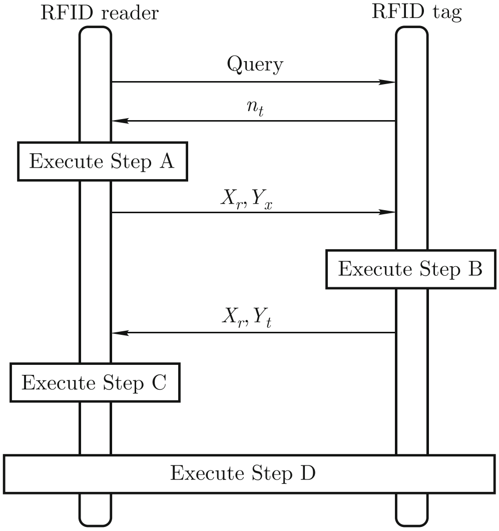

Fig. 10.6 Basic Access Control (BAC) for passport RFID tag.

substeps.

(1) Generate 2 random numbers, n_r and k_r.

(2) Compute $Z_r = n_r|n_t|k_r$

(3) Compute $X_r|Y_r$, where $X_r = E(Z_r, K_{\text{enc}})$ and $Y_r = h(X_r, K_{\text{mac}})$

After the tag receives $X_r|Y_r$, the tag will execute Step B. In this step, the tag will first verify Y_r using K_{mac}, and then decrypt X_r to obtain Z_r. The tag will check whether the n_t value in Z_r is the same as the value transmitted earlier. The tag will only continue executing if this n_t matches. The tag will finally compute $X_t|Y_t$ as follows.

(1) Generate random k_t.

(2) $Z_t = n_t|n_r|k_t$.

(3) Compute $X_t|Y_t$, where $X_t = E(Z_t, K_{\text{enc}})$ and $Y_t = h(X_t, K_{\text{mac}})$.

Upon receiving $X_t|Y_t$, the reader will execute Step C. Here, the reader will first verify that Y_t is valid using K_{mac}, and then decrypt X_t, and check whether the n_t value contained within Z_r is the same value it transmitted earlier.

Finally, in Step D, both the reader and tag will compute the session keys KS_{enc} and KS_{mac} using the value of K_r XOR K_t as seed. Subsequent communications between the reader and tag will be protected using KS_{enc} and KS_{mac}.

10.7.3 Security Analysis

A passport that only implements passive authentication does not meet the three security requirements. There is no access control mechanism, and thus any reader can query the tag and obtain the same information. Since the information returned by the passport is uniquely tied to the passport holder, passive authentication does not prevent illicit tracking. Finally, there is no protection against skimming. The adversary can simply query the tag, and then store the response from one passport onto another, to create a fraudulent passport.

The use of BAC can partly address these problems. Let us assume that the adversary does not have the information of K_{enc} and K_{mac}, i.e. the passport number, date of birth, expiry date of the passport is unknown to the adversary. The adversary querying the tag will be unable to get further than Step B (Fig. 10.6), since the adversary cannot return the X_r, Y_r values. Since all reader and tag communication is encrypted with K_{enc}, the adversary learns no information through eavesdropping. Illicit tracking requirement is satisfied because the tag returns a different n_t each time it is queried. Finally, since the adversary does not know K_{enc} and K_{mac}, a fake tag created by the adversary will be detected by a legitimate RFID reader.

Early work by Juels et al.[35] indicated the security pitfalls of implementing only passive authentication on passports, and advocated the use of BAC regardless of its limitations (early RFID enabled passports did not have BAC). More recent works have found practical security vulnerabilities for passports from specific countries[36]. BAC relies on the adversary being unaware of K_{enc} and K_{mac}, and work by Liu et al.[37] demonstrated such attacks. A practical tracking attack has been proposed by Chothia et al.[34] which can possibly be used to track passports based on the country of origin.

10.8 Conclusion

In this chapter, we studied the problem securing RFID networks and communications. The chapter focused on the weakest link, which is between the RFID reader and the RFID tag.

We described the characteristics of each of the components that make up an RFID network, and then categorized the security requirements for an RFID network. We first studied hardware based solutions to address these requirements. Then, we studied the more conventional protocol based approach towards RFID security. We studied protocols that can address the basic security requirements of preventing unauthorized access, illicit tracking, and skimming. We then turned our attention to security protocols that provide more advance security requirements of preventing mafia attack and providing grouping proofs. Finally, we concluded by studying a commercial RFID security protocol, the basic access control standard, used in passports.

We believe that RFID security will continue to be an important research area in the future as RFID is used in more critical applications. This chapter summarizes some of the key research in the security of RFID networks and communications, and we hope that our work can be used as a building block for future investigations into this problem.

References

[1] Gershenfeld N, Krikorian R, Cohen D (2004) The Internet of Things. Scientific American 291: 76 – 81.

[2] Koscher K, Juels A, Kohno T, Brajkovic V (2009) EPC RFID Tags in Security Applications: Passport Cards, Enhanced Drivers Licenses, and Beyond. In Proceedings of the 2009 ACM Conference on Computer and Communications Security (CCS'09), pp. 33 – 42.

[3] Kargoth G, Moskowitz P A (2005) Disabling RFID tags with visible confirmation: clipped tags are silenced. In Proceedings of the 2005 ACM workshop on Privacy in the electronic society (WPES'05), pp. 27 – 30.

[4] Moskowitz P A, Lauris A, Morris S (2007) A Privacy-Enhancing Radio Frequency Identification Tag: Implementation of the Clipped Tag. In Proceedings of the 2007 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom'07), pp. 348 – 351.

[5] Juels A, Rivest R, Szydlo M (2003) The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy. In Proceedings of the 2003 ACM Conference on Computer and Communication Security (CCS'03), pp. 103 – 111.

[6] Floerkemeier C, Roland S, Marc L (2004) Scanning with a Purpose — Supporting the Fair Information Principles in RFID Protocols. International Symposium on Ubiquitous Computing Systems.

[7] Rieback M R, Crispo B, Tanenbaum A S (2005) RFID Guardian: A Battery-Powered Mobile Device for RFID Privacy Management. Australasian Conference on Information Security and Privacy.

[8] Juels A, Ravikanth P (2003) Squealing Euros Privacy Protection in RFID-Enabled Banknotes. Financial Cryptography and Data Security, 2742: 103 – 121.

[9] Cai S, Li Y, Li T, Deng R H, Yao H (2010) Achieving High Security and Efficiency in RFID-tagged Supply Chains. International Journal of Applied Cryptography 2(1): 3 – 12.

[10] Avoine G (2011) RFID Security & Privacy Lounge. http://www.avoine.net/rfid/. Accessed 14 June, 2011.

[11] Alomair B, Poovendran R (2010) Privacy versus Scalability in Radio Frequency Identification Systems. Computer Communication, Elsevier, 33(18): 2155 – 2163.

[12] Tsudik G (2006) YA-TRAP: Yet Another Trivial RFID Authentication Protocol. International Conference on Pervasive Computing and Communication.

[13] Chatmon C, Tri van L, Burmester M (2006) Secure Anonymous RFID Authentication Protocols. Florida State University Technical Report. http://www.cs.fsu.edu/∼burmeste/TR-060112.pdf. Accessed 9 December, 2010.

[14] Tsudik G (2007) A Family of Dunces: Trivial RFID Identification and Authentication Protocols. In Proceedings of the 7th International Conference on Privacy Enhancing Technologies (PET'07), pp. 45 – 61.

[15] Hopper N J, Blum M (2001) "Secure Human Identification Protocols." International Conference on Theory and Application of Cryptology and Information Security (ASIACRYPT).

[16] Juels A, Weis S (2005) Authenticating Pervasive Devices with Human Protocols. Advances in Cryptology (CRYPTO).

[17] Piramuthu S (2006) HB and Related Lightweight Authentication Protocols for Secure RFID Tag/Reader Authentication. Collaborative Electronic Commerce Technology and Research.

[18] Holcom D, Burleson W, Fu K (2007). Initial SRAM state as a Fingerprint and Source of True Random Numbers for RFID Tags. Conference on RFID Security.

[19] Melia-Seguil J, Garcia-Alfaro J, Herrera-Joancomarti J (2010) Analysis and improvement of a pseudorandom number generator for EPC Gen2 tags. International Workshop on Lightweight Cryptography for Resource-Constrained Devices.

[20] Peris-Lopez P, Millan E S, van der Lubbe JCA, Entrena L A (2010) Cryptographically secure pseudo-random bit generator for RFID tags. International Conference for Internet Technology and Secured Transactions.

[21] Tan C C, Sheng B, Li Q (2010) Efficient techniques for monitoring missing RFID tags. IEEE Transactions on Wireless Communication 9(6): 1882 – 1889.

[22] Tan C C, Sheng B, Li Q (2008) Secure and Serverless RFID Authentication and Search Protocols. IEEE Transactions on Wireless Communication 7(4): 1400 – 1407.

[23] Hancke G P, Kuhn M (2005) An RFID Distance Bounding Protocol. Conference on Security and Privacy for Emerging Areas in Communication Networks.

[24] Avoine G, Muhammed A B, Suleyman K, Cédric L, Benjamin M (2010) A Framework for Analyzing RFID Distance Bounding Protocols. J. Comput. Secur. 19(2): 289 – 317.

[25] Kim C H, Avoine G (2009) RFID Distance Bounding Protocol with Mixed Challenges to Prevent Relay Attacks. International Conference on Cryptology And Network Security.

[26] Capkun S, El Defrawy K, Tsudik G (2010) GDB: Group Distance Bounding Protocols. arXiv.org, http://arxiv.org/abs/1011.5295. Accessed 15 January, 2011.

[27] Bolotnyy L, Gabriel R (2009) Generalized "Yoking-Proofs" and Inter-tag Communication. In Development and Implementation of RFID Technology. I-Tech Education and Publishing.

[28] Juels A (2004) "Yoking-Proofs" for RFID Tags. International Workshop on Pervasive Computing and Communication Security.

[29] Burmester M, de Medeiros B, Motta R (2008) Provably Secure Grouping-Proofs for RFID Tags. Smart Card Research and Advanced Applications (CARDIS).

[30] Garcia F, Koning G G, Muijrers R, Rossum P, Verdult R, Schreur R W, Jacobs B (2008) Dismantling MIFARE Classic. European Symposium on Research in Computer Security (ESORICS).

[31] Kasper T, Silbermann M, Paar C (2010) All You Can Eat or Breaking a Real-World Contactless Payment System. Financial Cryptography and Data Security. 6052: 343 – 350.

[32] Nohl K, Evans D, Starbug, Plotz H (2008) Reverse-Engineering a Cryptographic RFID Tag. USENIX Security Symposium.

[33] International Civil Aviation Organization (2009). Doc 9303 Part 1 Vol 1 and 2. http://www2.icao.int/en/MRTD/Pages/Downloads.aspx. Accessed 15 January 2011. Accessed 15 January, 2011.

[34] Chothia T, Smirnov V (2010) A Traceability Attack against e-Passports. Financial Cryptography and Data Security, 6052: 20 – 34.

[35] Juels A, Molnar D, Wagner D (2005) Security and Privacy Issues in E-passports. Conference on Security and Privacy for Emerging Areas in Communication Networks.

[36] Avoine G, Kassem K, Jean-Jacques Q (2008) ePassport: Securing International Contacts with Contactless Chips. Financial Cryptography and Data Security, 5143: 141 – 155.

[37] Liu Y, Kasper T, Lemke-Rust K, Paar C (2007) E-Passport: Cracking Basic Access Control Keys. OTM confederated international conference on the move to meaningful Internet systems: CoopIS, DOA, ODBASE, GADA, and IS - Volume Part Ⅱ.