Practical eavesdropping

of control data from EPC Gen2 queries with a programmable RFID toolkit

Like in many other emerging technologies, security threats can target the unprotected wireless channel used between RFID readers and tags to exchange information.

What you will learn…

- What is EPC Gen2
- How to use an eavesdropping attack to capture EPC Gen2 control sequences

What you should know…

- Radio Frequency Identification (RFID) protocol basics
- JAVA & Matlab programming

We present a practical eavesdropping attack to capture control data exchanged between a standard EPC Gen2 reader and a series of EPC Gen2 tags. We assume that the attacker can only access to the spectrum channel that contains the set of queries generated by the reader, that is, the reader-to-tag channel. We show that, even if the security model of the EPC Gen2 standard allows to capture this data, it contains information generated by the on-board components of the tags, that shall remain secret to guarantee the security properties of an EPC Gen2 network. We base our attack on a programmable RFID toolkit, that silently captures and stores reader interrogations.

Later, an analysis of such queries may be performed in order to retrieve the control data. The information provided in this article can be used to lead security analysts of EPC Gen2 deployments. Tests are conducted using standard EPC Gen2 messages. Captured data can be used in order to evaluate the statistical properties of the control sequences generated on-board of tagged items, as well as to characterize information such as the model and capabilities of the tags.

Brief Introduction to EPC Gen2

EPC Gen 2 is short-hand for the Electronic Product Code (EPC) Class-1 Generation-2 (Gen2) *Ultra-High Frequency* (UHF) *Radio Frequency Identification* (RFID) Protocol, the specification developed by EPCglobal for the second generation RFID air interface protocol and one example of a passive RFID tag protocol. For passive RFID we understand that the reader supplies the necessary energy for its proper performance to the tag, through radio-frequency signal. EPC Gen 2 was developed to establish a standard for RFID tags used in supply chain applications (e.g., tracking inventory). The current ratified standard for Class 1 devices operates in the UHF range (860 to 960 MHz), supports operation at long distances (e.g., 5-7 meters), and has minimal support for security (e.g., static passwords to access or kill information on the RFID device). The specification can be found at *http://www.epcglobalinc.org*.

Security Concerns

Even without physical access to the components of an EPC Gen2 network, an attacker can still try to get unauthorized access to data generated from tagged objects. This can be used later, to prepare further attacks, such as tag memory alteration, cloning, impersonation, and denial of service.

The EPC Gen2 specification only considers two basic on-board security features on the tag side:

- Pseudorandom Number Generators (PRNGs)
- Password-protected Operations

The pseudorandomness offered by on-board PRNGs is used in the following occasions in the EPC Gen2 communications protocol:

- Protect the password-protected operations
- As an anti-collision mechanism for inventorying processes
- To acknowledge other Gen2 specific operations such as memory writing, decommission of tags and self destruction

Figure 1 depicts an example of the EPC Gen2 communication protocol between a tag and reader, where pseudorandom control sequences are used for different issues. This example shows the necessary steps to modify specific areas of the tag's memory banks:

- Step 1: The reader sends a Query command to activate the tags in the surrounding area
- Step 2: The tag answers with a 16-bit sequence (hereinafter denoted as RN16) from its on-board PRNG
- Step 3: If the sequence is correctly received in the reader side, an acknowledgment is generated
- Step 4: Once the acknowledgement is received, the tag is ready to send its current 96-bit identification code (EPC). At this point, the communication has ended, except if the reader wants to access or modify the tag's memory banks
- Step 5: To do so, the reader requests a new RN16 from the PRNG of the tag
- Step 6: The new 16-bit sequence, denoted as Handle in Figure 1, is intended to act as session key for the subsequent operations
- Steps 7 to 14: These steps are only necessary if the tag is protected, that is, if a 32-bit password is enabled. In these steps, the reader requests two new RN16 control sequences, in order to encrypt the password (in two halves) with an XOR operation (denoted by the exclusive-or operator in Figure 1) through an Access command. Notice that these RN16 control sequences are previously sent in plaintext form from the tag
- Step 15: Using the Handle key, the reader requests a new RN16 sequence from the PRNG of the tag
- Step 16: Tag answers with the RN16 in plaintext form
- Step 17: The reader encrypts the information to be stored in the tag XORing the RN16 sequence, and writes it to the tag

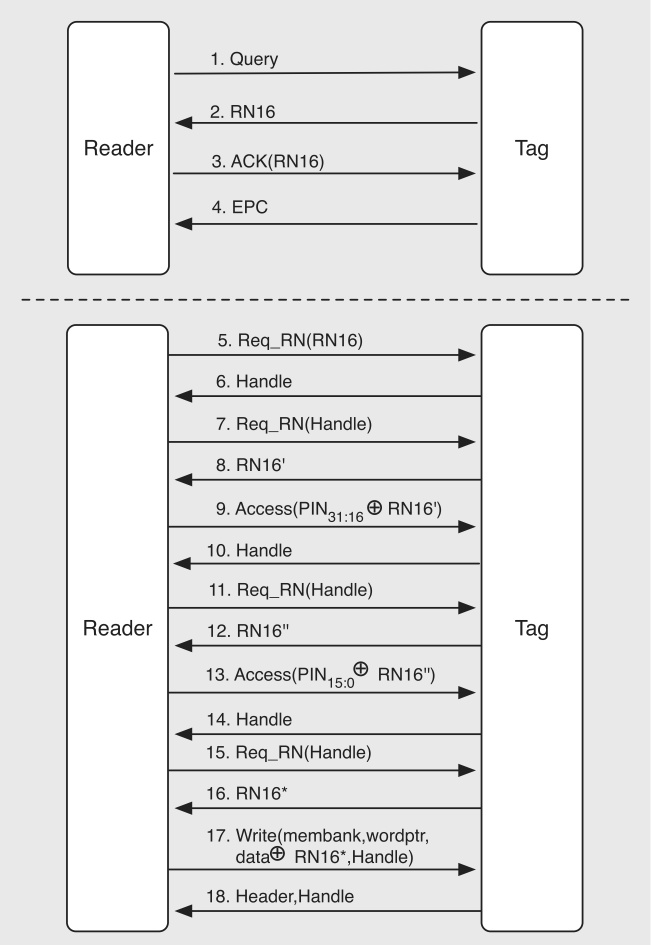


Figure 1. *Gen2 tag interrogation and memory writing*

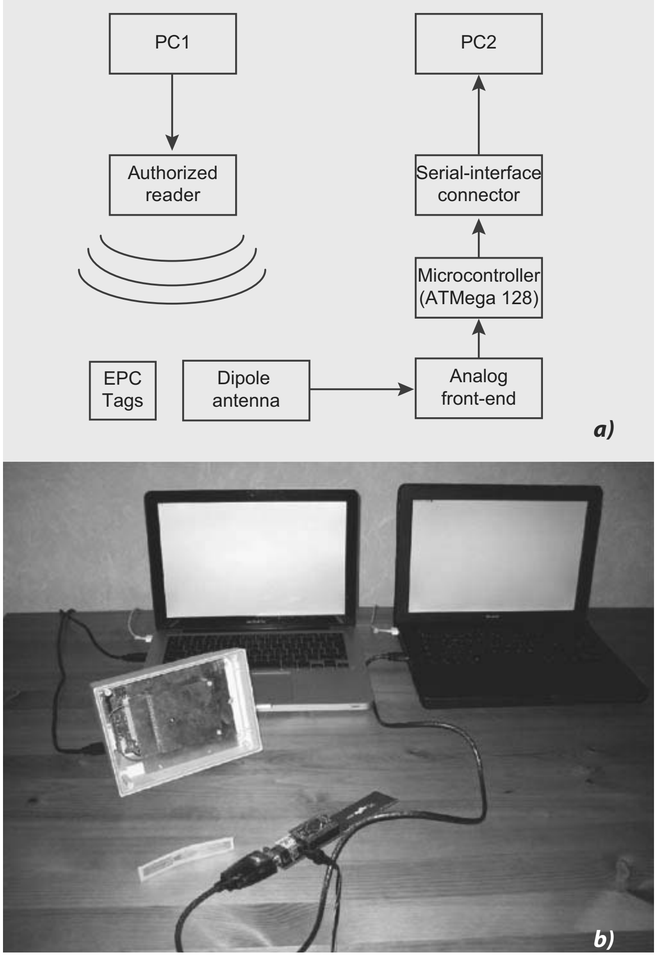


Figure 2. *Proposed setup*

Figure 3. *CAEN A829EU Reader*

- Step 18: If the tag's memory bank is successfully modified, the tag confirms the operation returning the Handle to the reader. Steps 15 to 18 are repeated until the desired memory banks are modified

Let us observe from our previous example that, even if it might be difficult for an attacker to retrieve the data associated to password protected operations, it is theoretically possible to be done. In our example, it suffices to intercept (or predict) the RN16' and RN16" control sequences used by the reader to encrypt the password, in steps 8 and 12, and simply apply an XOR operation to the contents of steps 9 and 13. Although the security model of the EPC Gen2 specification does not assume an adversary capable of eavesdropping the tag-to-reader channel, statistical analysis of the interrogations of an EPC Gen2 reader (that is, the reader-to-tag channel) may help. Indeed, statistical analysis of the reader interrogations may lead the eavesdropper to characterize the exchange of information, with the goal of predicting those outgoing control sequences provided by the tag. Eventually, this prediction may allow to bypass the security of the remainder password-protected commands defined in the Gen2 standard. It is, therefore, very important to assess the feasibility of this threat. In the sequel, we show a practical setup that might help to do so.

Proposed Setup

Figure 2 shows our experimental setup (diagram and real deployment). In this deployment, an authorized reader is communicating with real EPC Gen2 tags, while a capture device is eavesdropping the exchanged information. This experimental setup contains the following devices:

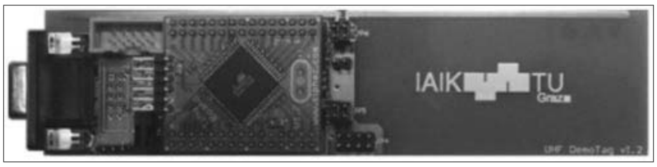


Figure 4. *IAIK UHF Demo Tag*

EPC Gen2 Reader

We used a CAEN A829EU, compliant with ISO 18000-6C and the EPC Gen2 standards, which comes with all the necessary tools to develop new front- and back-end applications (Figure 3).

EPC Gen2 Tags

Commercially available ISO 18000-6C and the EPC Gen2 tags for item-level tracking.

Capture device

The IAIK UHF Demo Tag is a programmable device intended for developing new commands or functionalities to the EPC Gen2 standard. Here it is used to store the reader-to-tag commands (Figure 4).

Two laptops

Used to manage the CAEN A829EU reader and the IAIK UHF Demo Tag

The IAIK UHF Demo Tag consists on four main components: an antenna, a *radio-frequency* (RF) front-end, a programmable microcontroller, and a firmware library. The IAIK UHF Demo Tag allows the verification of new functionality using compliant EPC Gen2 readers, by modifying the code inserted into the Demo Tag. Thus, new developments can be implemented and tested in

Product Name	Characteristics	Layout
UPM ShortDipole^{x}	*Inlay Manufacturer:* UPM Raflatac *Integrated Circuit:* NXP G2XL *Antenna size:* 92 x 11 mm *Die-cut size:* 97 x 15 mm *Applications:* – logistics – item-level tracking	
UPM DogBone	*Inlay Manufacturer:* UPM Raflatac *Integrated Circuit:* Impinj Monza 3 *Antenna size:* 92 x 8 mm *Die-cut size:* 105 x 12 mm *Applications:* – logistics	
Confidex Cassey™	*Inlay Manufacturer:* Confidex *Integrated Circuit:* NXP G2XL *Antenna size:* 88 x 22 mm *Die-cut size:* 92 x 25 mm *Applications:* – logistics – item-level tracking	
Alien Squiggle	*Inlay Manufacturer:* Alien Technology *Integrated Circuit:* Alien Higgs 3 *Antenna size:* 95 x 8 mm *Die-cut size:* 99 x 13 mm *Applications:* – logistics – item-level tracking	
Trace Inlay	*Inlay Manufacturer:* Trace Tecnologias *Integrated Circuit:* Impinj Monza 3 *Antenna size:* 97 x 14 mm *Die-cut size:* 108 x 24 mm *Applications:* – logistics	

Figure 5. *Sample commercial tags used on our setup*

real environments. The programmable microcontroller of the IAIK UHF Demo Tag consists of an Atmel AVR ATmega128. An implementation of the EPC Gen2 protocol for the ATmega128 is stored as a firmware library in the flash memory of the microcontroller. This library contains all the necessary functions to collect and pre-process standard EPC Gen2 queries and responses, as well as to visualize the memory buffer that contains the reader-to-tag captured messages. The microcontroller is connected to the second laptop via an UART module and, in turn, to a serial-interface connector. This serial interface allows us to interact with the capture device. It provides basic operations such as memory mapping, EPC Gen2 values' configuration, visualization of the captured queries, modification of the queries, as well as a wide range of user defined operations.

With regard to the set of EPC Gen2 tags, we show in Figure 5 the layout and characteristics of some of the tags used in our setup. All the tags are ISO 18000-6C and EPC Gen2 compliant, and commercially available for their use in real-world applications, such as logistics and item-level tracking. Logistics includes a wide range of applications, being the most relevant applications for supply chain management, industrial asset management, airline baggage handling, and tracking of cases and pallets (e.g., WalMart, Department of Defense, and METRO Group tracking applications). Item-level tracking includes applications for sectors such as pharmaceuticals, healthcare, and libraries. Some other applications based on these tags can also consist on brand protection and anti-counterfeiting.

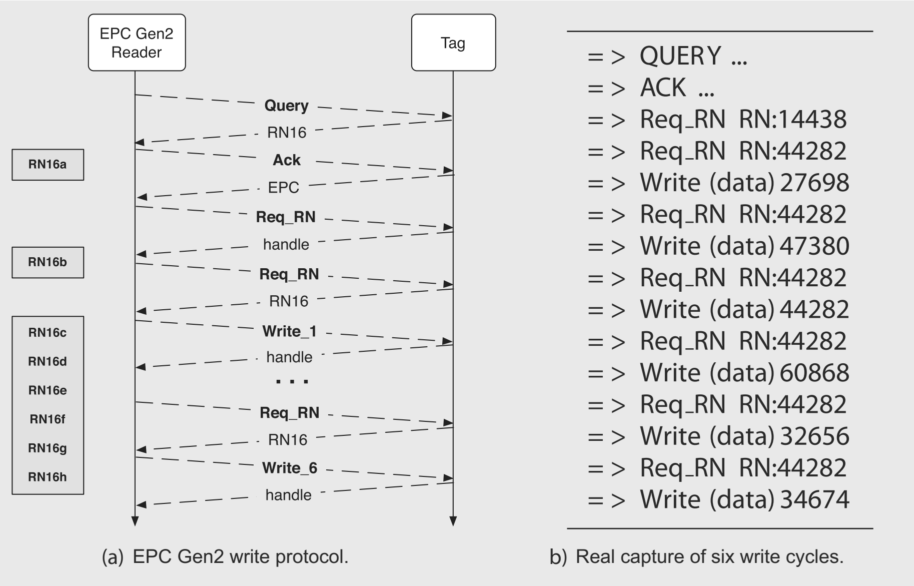
The tags are all pre-programmed with a 32-bit access and kill passwords to grant access for reading or writing their internal user nonvolatile memory. Regarding the latter capability, the memory space ranges from 96 to 496 bits in most cases. Observe that different models from the same vendor, or even different products commercialized by different vendors, often share the same *integrated circuits* (ICs). Indeed, EPC Gen2 *Integrated Circuit* (IC) manufacturers may directly commercialize their ICs and inlay together (that is, the tag ready to work), or as a single product to add to the inlays of other vendors. Hence, there are several different tags sharing the same IC. For this reason, we refer to the tags using their IC model, rather than the tag model or product itself. Hence, the results shown in the next section group the series of tags based on their specific EPC Gen2 inlays, which are: NXP G2XL, Alien Higgs 3, and Impinj Monza 3.

Execution and Sample Results

The goal of our eavesdropping attack is to capture the pseudorandom (RN16) control sequences from standard EPC Gen2 queries. To increase the odds of capturing more control sequences, we check the amount of RN16s used on the standard EPC Gen2 operations. The higher the number of RN16s per operation, the higher the amount of eavesdropped sequences. Table 1 shows the mandatory operations for Gen2 reader-to-tag protocol and the minimum number of RN16s involved in each operation. Note that the write command generates a minimum of eight RN16s for its proper execution. For instance, assuming a full EPC code (96 bits) memory writing operation, up to six RN16s must be generated to cover the reader-to-tag communication, besides the two previously generated pseudorandom sequences for the inventory query and the corresponding descriptors. If the tag is password-protected, two additional RN16s are generated to cover the 32-bit password.

Table 1. *Minimum number of RN16s involved in EPC Gen2 operations*

Operation	Inventory	Access			
Command	Identification	Read	Write	Lock	Kill
Number of RN16s	1	2	8	2	4



(a) EPC Gen2 write protocol.

```
= > QUERY ...
= > ACK ...
= > Req_RN  RN:14438
= > Req_RN  RN:44282
= > Write (data) 27698
= > Req_RN  RN:44282
= > Write (data) 47380
= > Req_RN  RN:44282
= > Write (data) 44282
= > Req_RN  RN:44282
= > Write (data) 60868
= > Req_RN  RN:44282
= > Write (data) 32656
= > Req_RN  RN:44282
= > Write (data) 34674
```

b) Real capture of six write cycles.

Figure 6. *Write process for EPC Gen2 and the PRNG utilization. In (a), we can see the six cycles of the EPC Gen2 write command. In (b), we can see a real sample of six write cycles captured from the reader-to-tag channel*

The EPC Gen2 communication protocol bases its security in the different channels strength (as a

Glossary

ELECTRONIC PRODUCT CODE (EPC). Family of coding schemes defined by EPCGLOBAL Inc.

EPCGLOBAL Inc. Joint venture between GS1 (Global Standards One, formerly known as EAN International) and GS1 US™ (formerly the Uniform Code Council, Inc.) created to commercialize the EPC technology.

EPC NUMBER. A tag data format compatible with the family of coding schemes proposed by EPCglobal Inc. It typically contains: a header, pointing out the family code that is being used; a manufacturer code; an object class; and a serial number.

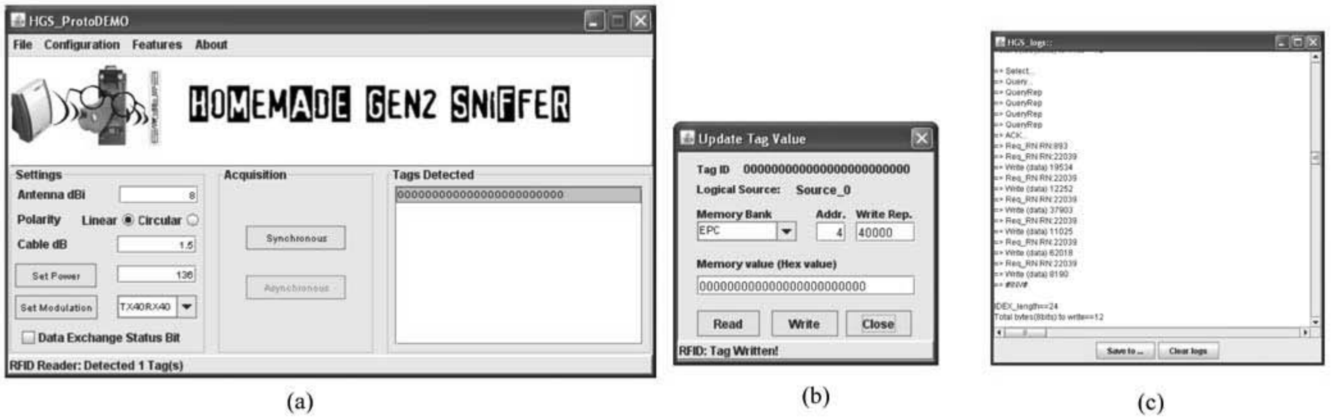
EPC NETWORK. A service oriented architecture defined by EPCglobal Inc. that proposes the integration of RFID and Internet technologies to enable automatic identification and sharing of item data in supply chain applications.

PASSIVE TAG. RFID component attached to an item. It contains information about the item. Since it does not have its own power source, it provides the information by backscattering a reader's signal.

PSEUDORANDOM NUMBER GENERATOR (PRNG). Algorithmic solution to generate deterministic sequences of binary strings.

READER. RFID component that requests and receives information from tagged items.

TAG IDENTIFICATION (TID). Memory bank or identifier that uniquely identifies a RF tag.



Antenna dBi 8

Polarity Linear ● Circular ○

Cable dB 1.5

Set Power 136

Set Modulation TX40RX40

□ Data Exchange Status Bit

000000000000000000000000

RFID Reader: Detected 1 Tag(s)

(a)

Tag ID 000000000000000000000000

Logical Source: Source_0

Memory Bank EPC

Addr. 4

Write Rep. 40000

Memory value (Hex value) 000000000000000000000000

RFID: Tag Written!

(b)

```
=> Select
=> Query
=> QueryRep
=> QueryRep
=> QueryRep
=> QueryRep
=> ACK
=> Req_RN RN:893
=> Req_RN RN:22039
=> Write (data) 19534
=> Req_RN RN:22039
=> Write (data) 12252
=> Req_RN RN:22039
=> Write (data) 37903
=> Req_RN RN:22039
=> Write (data) 11025
=> Req_RN RN:22039
=> Write (data) 62018
=> Req_RN RN:22039
=> Write (data) 8190
=> #RN#

IDEX_length==24
Total bytes(8bits) to write==12
```

(c)

Figure 7. *Sample execution of our proposed attack*

result of the power supply method for passive RFID). Pseudorandom control sequences are sent in plaintext form from tag to reader with a weak signal (5-7 meters), but reader to tag information is sent through a powerful signal which can be received to a thousand meters. We expect to capture the RN16 sequences by eavesdropping the reader-to-tag channel, to be consistent with the security model defined by EPCglobal in the EPC Gen2 standard. Since reader-to-tag messages include the acknowledgment of the control sequences that are computed from the on-board *pseudorandom number generator* (PRNG) included on the EPC tags, we can, therefore, capture those sequences by simply collecting the reader acknowledgments.

As we already covered in the *Security Concerns* section, an EPC Gen2 write operation is an access command used to modify specific areas of the tag memory. Regardless of its open or secured state, a minimum of 128 bits are generated from the tag's PRNG, included in the several write necessary query-response commands to write a full EPC code (generally, 96 bits).

Based on these observations, we can construct a database of control sequences for each one of the proposed categories of tags in our testbed by iteratively executing predefined series of write challenges towards the tags. Each challenge that is sent towards a tag provides to our capture device 128 bits of additional pseudorandom control data. Figure 6(b) shows a simple example where

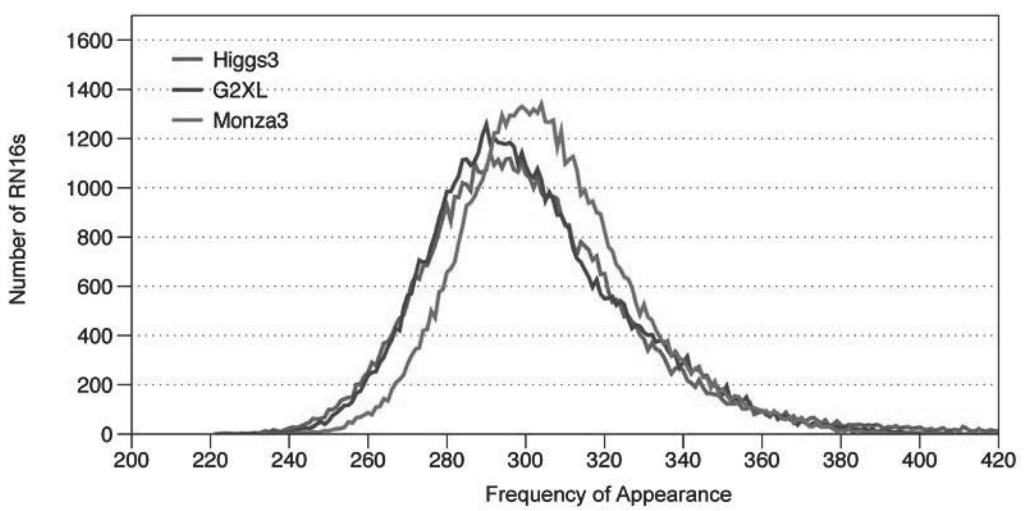


Figure 8. *Sample results after applying some statistical processing to the captured data*

On the Web

- *http://jce.iaik.tugraz.at/sic/Products/RFID-Components* – IAIK UHF Demo Tag toolkit technical specifications
- *http://www.caenrfid.it/rfid/products.php* – CAEN A829EU technical specifications
- *http://www.youtube.com/watch?v=DzwdCimLrDI* – video showing the execution of the technique presented in this article

six write cycles are captured. These captures allow us to collect 96 pseudorandom control bits generated from the on-board PRNG of a given EPC Gen2 tag. Later on, and via the serial output interface of the RFID toolkit, these sequences will be parsed and translated with standard Matlab routines, in order to post-process and count their frequency of appearance. Figure 6(b) shows the reader-to-tag challenges. The reader sets the EPC identification to 0 (that is, $EPC_{96b} = 0_{i\ b\ldots(i+16)\ b}$ XOR RN16), to obtain the series of pseudorandom sequences directly from the data field of the *write* challenges.

Figure 7 shows the execution of our eavesdropping attack. It is based on screen captures of the single front-end GUI application adapted from the Java sample tools provided with the CAEN A829EU reader, and that allows us to centralize also the automatic storage of control sequences captured during the attack. In (a), we can see the main interface of our adapted front-end. In (b), we can see the Update Tag Value screen ready to perform 40,000 write commands. In (c), we can see the contents of the IAIK Graz UHF Demo Tag Buffer, connected via its serial device to the GUI of our front-end application, and showing one write sequence eavesdropped from the reader-to-tag channel.

Finally, using standard transformations based on Matlab code, it is straightforward to proceed with the execution of statistical processing of the captured data. For instance, to analyze and identify the source of the sequences collected by the IAIK Demo Tag of our setup. Figure 8 shows an example, in which we simply represent the frequency of occurrence of each of the sets of control sequences collected from every type of tags used in our sample setup. More sophisticated processing and attacks may lead the eavesdropper to the following steps of a more elaborated attack. This will be presented in a future article.

Conclusion

Existing commercial products in the market can be used to test some of the security deficiencies in today's RFID technology. In this article, we have targeted the EPC Gen2 case, which is an international standard that proposes the use of *Ultra High Frequency* (UHF) RFID protocols for item-level tracking. EPC Gen2 tags are designed to balance cost and functionality. As a consequence, security on board of Gen2 tags is minimal. It is, indeed, mainly based on the use of on-board pseudorandom number generators (PRNGs), used to obscure the communication between readers and tags; and to acknowledge the proper execution of password-protected operations. We have presented a practical setup that shows how simple is to capture and analyze the pseudorandom control sequences generated by real world EPC Gen2 tags. Indeed, we have seen that these sequences are disclosed over the unprotected wireless reader-to-tag channel and, therefore, easy to capture even at long distances. The capture and analysis of these sequences should be followed by an analysis of their statistical properties. Why? because if a deviation or characterization of the on-board generators that created those sequences is discovered, a passive attacker could try later to perpetrate some more complex attacks, such as unauthorized execution of password-protected operations. We hope that this practical example may succeed at encouraging you to keep exploring all the fascinating possibilities that the RFID technology may still offer us; while, at the same time, making you more conscious of the security threats that this technology must handle before letting it invading our future lives.

Acknowledgements

We would like to thank the editors for the fruitful discussions and support. We would also like to acknowledge support by the Spanish Ministry of Science and Innovation, w.r.t grants TSI2007-65406-C03-03 E-AEGIS and CONSOLIDER CSD2007-00004 ARES.

JOAQUIN GARCIA-ALFARO

Joaquin Garcia-Alfaro is associate researcher at Telecom Bretagne, Institut Telecom (Rennes, France). joaquin.garcia-alfaro@acm.org.

JORDI HERRERA-JOANCOMARTI

Jordi Herrera-Joancomarti is associate professor at the Universitat Autonoma de Barcelona (Bellaterra, Spain). jordi.herrera@uab.cat

JOAN MELIA-SEGUI

Joan Melia-Segui is postdoctoral researcher at the Universitat Oberta de Catalunya (Barcelona, Spain). melia@uoc.edu