A Proposal for an Authentication Protocol in a Security Layer for RFID Smart Tags

Martin Feldhofer

Institute for Applied Information Processing and Communications (IAIK)
Graz University of Technology
Inffeldgasse 16a, 8010 Graz, Austria

Abstract

This article presents a proposal for an authentication protocol for Radio Frequency Identification (RFID) smart tags. RFID tags are microchips attached to products to identify them contactless during production or in use via radio frequency. Cryptographic authentication is necessary to protect branded goods from forgery. Existing protocols do not include cryptographic authentication mechanisms. Therefore, a new approach for authentication is proposed in this paper. Because of the limited computing power, low die-size, and low-power requirements a two-way challenge-response authentication scheme is used. Packet and frame formats are presented to include the new approach to the existing protocol which is defined in the ISO/IEC 18000 standard. To verify this approach Java models in different abstraction levels were implemented. The hardware implementation was done in VHDL for an FPGA target device to get a fast prototype.

Keywords: RFID, radio frequency identification, smart tag, reader, authentication protocol, challenge-response, ISO/IEC 18000, FPGA implementation.

1 Introduction

The need for identification of various products and goods increases in our automated world. Every today's business products must be identified during its way from producer to consumer, or in use, a lot of times [3]. Using

an *Radio Frequency Identification System (RFID)* is a good approach for automated identification of products. RFID (smart) tags are microchips attached to everyday products to identify them. The ISO/IEC 18000 standard defines a protocol for RFID tags that handles bi-directional communication between a reader device and an RFID tag [4]. Unfortunately, there are no mechanisms defined to authenticate a tag to the reader. This is necessary when a manufacturer wants to protect its branded products from plagiarism or a customer wants to be sure that his or her article is produced by that company it claims to be. The best way to implement authentication on RFID tags is to add cryptographic algorithms. Today, there are no implementations of RFID tags with strong cryptographical authentication included because of the hard requirements concerning low die-size and low-power consumption.

Another aspect is the compatibility to the existing standards. This is important to reduce the effort for production of new reader devices. Therefore, the existing standard should be expanded. Public-key cryptography with a three-way challenge-response protocol would be the best solution from the security point of view. Because of the high computation effort this is not possible on RFID tags. Instead of that a practical approach with symmetric cryptography and a two-way handshake mechanism is shown in this paper.

The remainder of this article is structured as follows. Section 2 explains the basics of RFID systems and section 3 shows the ISO/IEC 18000 standard where the communication protocol principles are defined. Section 4 briefly describes the background of authentication mechanisms which are the basics for the proposed authentication protocol in section 5. A possible implementation approach is shown in section 6. Section 7 presents the results and conclusions are drawn in section 8.

2 Radio Frequency Identification Systems

The automated identification of products is often necessary during the life cycle of a product. There exist a lot of different methods for identification but in the last few years the main attention is focused on systems using radio frequency. Barcode systems are very simple to use but have the disadvantage that the amount of data that can be stored is not very large. It is also impossible to change the value on a barcode label.

A solution to this inflexibility is the usage of microchips. In addition to already known smart cards that need mechanical contact to the reader, wireless devices where data and energy is transmitted via radio frequency are used. Such systems are called *Radio Frequency Identification Systems (RFID).*

RFID systems are similar to smart cards. Data can be stored and processed on the chip. The energy supply and the data transfer are done wirelessly via an electromagnetic field. There is no need for physical contact nor of a line of clear sight between the device and the reader. Because of this advantage, RFID devices get more and more important. The three major areas of application are transportation and distribution, manufacturing and processing, and security.

2.1 Characteristics of RFID Systems

RFID systems always consist of two major components shown in figure 1:

- *Reader* including antenna which communicates with the tag,
- *Tag* or *Transponder* which is placed on the object to be identified.

The communication between them uses a defined radio frequency and protocol where the following three parts must be transferred. See figure 1.

- *Data* in both directions,
- *Clock* signal from reader to the tag,
- *Energy* from reader to the transponder to activate it.

The clock signal is recovered from the carrier signal by the transponder. Since passive transponders have no power supply on the chip, energy must also be transmitted from the reader to the tag.

2.2 Data Transmission

Passive tags that do not contain power supply and clock generation on the chip communicate in half-duplex mode with the reader. The following sections describe mechanisms for communication in both directions.

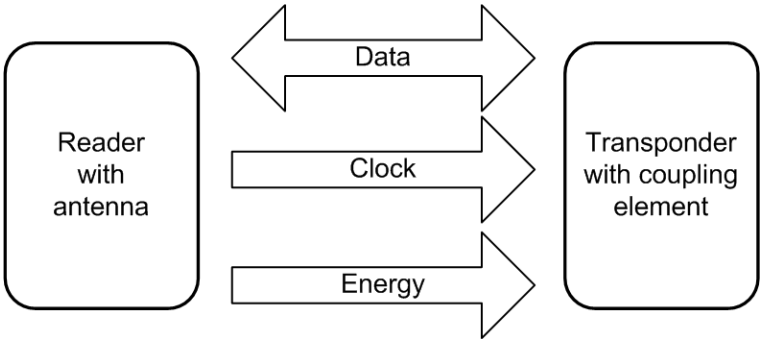

Figure 1: Components of an RFID system.

2.2.1 Reader-to-transponder communication

Data transmission from the reader to the transponder works with digital modulation. *Amplitude Shift Keying (ASK)* is used mostly because of its simple demodulation mechanism.

There exist two different ASK levels which can be used. Using 100%-ASK the carrier is switched on and off depending on the data to be sent. More often the 10% level is used because otherwise, there would be no power supply for the transponder during the off-phase of the carrier.

2.2.2 Transponder-to-reader communication

The non-existing power supply of the transponder requires a method where the carrier signal which is sent by the reader is also used for transponder-to-reader communication. That means that the transponder uses the energy of the operating field to return its response. This method is called *load modulation*. Load modulation is a mechanism where a load resistance is included in the circuit which is switched on and off depending on the data. This additional power consumption is recognized by the reader which detects zeros and ones in that way.

3 Protocol Definitions in the ISO/IEC 18000 Standard

The ISO/IEC 18000-3 standard describes the communication of RFID tags with a reader device using a frequency of 13.56 MHz. Modulation, framing, anti-collision methods, protocol parameters, and other specific information are presented in [4] and will be shortly described in the following.

3.1 Description of Communication Method

As described in section 2 communication between the reader and the transponder works via modulation. The reader uses ASK modulation with the two modulation indices 10% and 100%. Data coding is possible with "1 out of 256" or "1 out of 4" data coding where either one byte or two bits are encoded. The frame delimiters are implemented using code violation which means that Start-of-frame (SOF) and End-of-frame (EOF) delimiter look different from data.

The tag uses load modulation to send its response. Different modes concerning subcarrier and data rates are possible. The configuration is done via the application protocol. Data shall be encoded using Manchester coding to ease collision detection for the reader.

3.2 Overall Protocol Description

The transmission protocol defines how to exchange instructions and data between the reader and the transponder in both directions. It is based on the concept of "reader talks first" which means that any tag shall not start transmitting unless it has received and properly decoded an instruction sent by the reader. Every command consists of a request from the reader to the tag and a response from the tag to the reader. Requests and responses are contained within a frame with the delimiters SOF and EOF. The protocol is bit-oriented and the number of bits transmitted in a frame is a multiple of eight. Each request and response consists of the fields:

Flags: Indicate whether one or two sub-carrier frequencies and what data rate should be used for the response. Additional information is presented to address appropriated tags. Responses of the tag use the flags to indicate errors during transmission.

Command code: A one-byte constant that indicates which request is sent. There exist three major types of commands. Mandatory commands must be implemented by the tag. Optional ones could be implemented by the tag if they are necessary for the application. Custom commands can be used by manufacturers to include their own commands in the protocol. Custom commands are explained in detail in section 5 where the security layer implementation is shown.

Parameters and data fields: This are command specific data that include relevant information for processing a request and a response, respectively.

CRC: The cyclic redundancy check (CRC) is calculated on all bytes after the SOF up to but not including the CRC field. It is used to detect errors during transmission.

Depending on the command a request can be addressed or unaddressed. Each tag has to store an *unique identifier (UID)* that is used to address a specific tag. This number is 64-bit long and contains a manufacturer code and tag-specific data. It has to be unique for the whole world.

3.3 Anti-collision Mechanism

If more than one tag is within the environment of a reader, an anti-collision mechanism is needed. This is done via the mandatory *inventory request* of the reader. If the reader sends an unaddressed inventory request, all tags within the reception area try to send their answer. This answer is a response frame with the UID of the tag.

When two or more tags send a response, a collision occurs at the reader. This collision is detected by the reader and leads to a repetition of the request by adding a part of the UID, the so called mask value, to the request. The mask value can be up to 64 bits (which is the complete UID). The mask length is also included in the request and each tag must compare as many bits of its own UID with the mask value as the mask length indicates. When the mask length is chosen appropriately only one transponder should answer to a request. From this moment on the complete UID of the tag is known and each request could be addressed for this tag individually. The mechanism used to select a tag by the mask value and the mask length is called *binary-tree algorithm*. Identification of the tag to reader is also done

via the anti-collision mechanism because the reader obtains the UID from the tag.

4 Authentication

Authentication is assurance of the identity of an entity at the other end of a communication channel. There exist several methods concerning strong authentication. The main difference consists whether *secret-key* or *public-key* cryptography is used. In secret-key cryptography the signer and the verifier must share a secret where the problem of the key exchange must be solved. In public-key cryptography this problem does not exist because the private key is kept secret in the signer's environment and the public key is published with a certificate. The method using public-key cryptography is known as a *digital signature*. The protocols used for authentication are called *zero-knowledge protocols* and *challenge-response protocols*. The latter ones are used for RFID authentication in this proposal and work as explained in the following sections.

4.1 Challenge-response Protocol

In challenge-response protocols the verifier sends a challenge request to the claimant. This challenge can be a randomly chosen number which varies from one request to the other. The claimant "proves" its identity by manipulating the challenge using the secret which is associated with that entity. It is important not to show this secret to the verifier during the communication. After receiving the response from the claimant the verifier validates the response and can be sure whether the claimant knows the secret. When an attacker observes the communication between the verifier and the claimant it should not provide any information for a subsequent identification because the next challenge will be a different number.

4.1.1 Challenge-response by secret-key and public-key techniques

When using public-key cryptography, the verifier sends a challenge to the signer. This number is encrypted by the signer using the private key and sent back to the verifier who decrypts the response with the public key. When the result is the same as the verifier sent to the signer, he or she can be sure that the signer is the entity he or she claims to be. The advantage is that the signer needs not to show its private key to anybody else. The verifier just takes the public key out of a repository and can verify the signature.

Such a protocol can also be implemented using symmetric cryptography. Instead of decrypting the response from the signer with the public key, the verifier uses the shared secret key to decrypt the response and also compares the result with the sent request. The disadvantage of the key distribution problem must be taken into account, but most secret-key algorithms work faster than public-key methods.

4.1.2 Unilateral and mutual authentication

The protocol where one entity A is authenticated to entity B is called *unilateral authentication*. Thereby *one-way* and *two-way* challenge-response protocols are used. When using a one-way protocol a timestamp mechanism is needed. The signer A sends the encrypted timestamp t_A to the claimant B who decrypts it and verifies that the timestamp is acceptable. See equation 1. The two-way protocol works using random numbers. The claimant B must first send a random number r_B to the signer A who encrypts it and sends it back. Verification works by decrypting the response and comparing it with the random number sent. This protocol can be seen in equation 2.

$$A \rightarrow B: \quad E_K(t_A) \tag{1}$$

$$\begin{aligned} A \leftarrow B: &\quad r_B \\ A \rightarrow B: &\quad E_K(r_B) \end{aligned} \tag{2}$$

If both entities want to authenticate each other *(mutual authentication)*, a *two-way* or a *three-way* challenge-response protocol is used. For two-way authentication, see equation 3. The entity A must encrypt the timestamp t_A and a randomly selected number r_A and send it to the second party B. Decrypting and verifying the timestamp authenticates the party A to the entity B. Then, the random number r_A is encrypted and sent back to the originator who can decrypt the message and compare the result with the sent random number. A three-way mechanism as shown in equation 4 works similarly, but one additional transmission has to be done. Entity A chooses a random number r_A and sends it to B. This number r_A and another random number r_B are encrypted by B. The encrypted value is sent back

to A who verifies its own random number and encrypts the random number of B. This is sent to A who finally encrypts the number and compares it with its chosen number r_B.

$$\begin{aligned} A \rightarrow B: &\quad E_K(t_A, r_A) \\ A \leftarrow B: &\quad E_K(r_A) \end{aligned} \tag{3}$$

$$\begin{aligned} A \rightarrow B: &\quad r_A \\ A \leftarrow B: &\quad E_K(r_A, r_B) \\ A \rightarrow B: &\quad r_B \end{aligned} \tag{4}$$

5 Security Layer for RFID Tags

When implementing security issues for RFID tags attention should be paid to some special topics. Especially the limited computing power on one hand and the low die-size and low-power requirements on the other hand must be considered. That leads us to the need for efficient hardware and software implementations. Additionally, it is important to be compatible to existing standards like the ISO/IEC 18000 standard [4] for RFID tags operating at a frequency of 13.56 MHz. As security measures are not implemented until now, this paper describes a new method for implementing strong authentication of the tag to the reader.

5.1 Authentication Protocol

For meeting the above requirements a simple two-way challenge-response protocol is chosen for authentication. As described in section 3, the two-way protocol ideally fits to the overall communication structure where the reader sends a request and the tag responses. To include this command to the ISO/IEC standard, a new custom command is specified. A detailed view of the protocol frame format can be seen in section 5.2.

The authentication protocol is a *Simple Authentication and Security Layer (SASL)* protocol as specified in RFC 2222 [10]. Ideas and concepts are out of FIPS 196 [11] and from ISO/IEC standard 9798 [1]. For now, symmetric authentication algorithms are proposed because of their better performance. Using public-key algorithms should be more secure because the problem of

key exchange does not exist. For a detailed view of the protocol see figure 2. A 128-bit random number r_R is generated at the reader and sent to the tag within a request frame. The tag encrypts this random number and sends it back to the reader within a response frame. The reader must decrypt the received data and compares it with the sent data. If they are equal the reader can believe the authenticity of the tag.

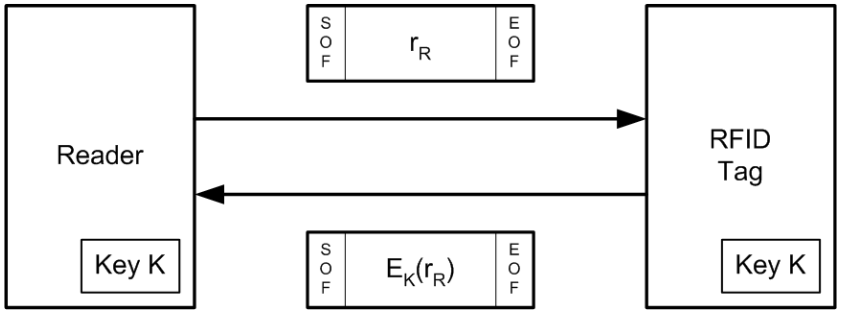

Figure 2: Proposed authentication protocol implementation.

5.2 Protocol Frame Format

As mentioned above frames are sent from reader to tag and vice versa. Such frames always are limited with special delimiters called *End-of-frame (EOF)* and *Start-of-frame (SOF)*. Within the frame data are organized in bytes. The *application protocol data units (APDUs)* for the authentication request and response can be seen in figures 3 and 4 in sections 5.2.1 and 5.2.2.

5.2.1 Request frame format

The request frame shown in figure 3 starts with an SOF delimiter followed by 8 bits of flags which are explained in section 3. The following byte is the command code. It is defined with "0xA0" meaning that this command is a custom command implemented by a manufacturer. Any custom command contains as its first parameter the IC manufacturer code (IC Mfg code). This allows IC manufacturers to implement custom commands without risking duplication of command codes and thus misinterpretation. The next parameter is the 64-bit unique identifier (UID) which addresses a sole

tag to answer to that request. This UID must first be retrieved from the tag by the inventory request. Before the 16-bit CRC value which is calculated on all bytes after the SOF up to but not including the CRC field the 128-bit challenge is included. The challenge should be a random number to prevent replay attacks. The frame is completed with the EOF delimiter.

SOF	Flags	0xA0	IC Mfg code	UID	Random number r_R	CRC	EOF
	8 bit	8 bit	8 bit	64 bit	128 bit	16 bit	

Figure 3: APDU for authentication request.

5.2.2 Response frame format

The response frame as shown in figure 4 also starts with the SOF delimiter followed by 8 bits of flags indicating an error during transmission. The 64-bit UID is included to identify the tag sending the response. After the 128 bits signed data the CRC follows. The ending delimiter is again the EOF.

SOF	Flags	UID	Signed data $E_K(r_R)$	CRC	EOF
	8 bit	64 bit	128 bit	16 bit	

Figure 4: APDU for authentication response.

5.3 Cryptographic Algorithm

The cryptographic algorithm proposed for the authentication protocol is the *Advanced Encryption Standard (AES)* [**?**]. The AES is the successor to the *Data Encryption Standard (DES)* and supports key sizes of 128 bits, 192 bits, and 256 bits, in contrast to the 56-bit keys offered by DES. The block size of 128 bits is equal to the length of the challenge from the reader. In the

128-bit key size version, AES consists of ten rounds, and in each round the individual bytes are transformed, the rows are rotated, and the columns are multiplied to a constant matrix. Each round is concluded with an XORing of the resulting array to a round key. For the authentication protocol the minimum key size of 128 bits is used. In addition to the small key size a minimalist AES hardware implementation is needed to reduce the chip area.

6 Protocol Implementation Considerations

Implementing the protocol in the ISO/IEC 18000 standard with the above extensions for authentication requires some considerations concerning the architecture of an RFID tag. These considerations can be split up in hardware and software requirements explained in the following. The requirements are explained in general independent of the chosen implementation. For example, software does not automatically mean written code. It is also possible to implement software hardcoded in so called "state machines".

6.1 Software Requirements

The main components of software are the controlling functions. The implementation of the commands in the ISO/IEC protocol needs algorithms for anti-collision detection and CRC generation. The authentication protocol must implement a cryptographic algorithm and needs to handle the response of the challenge from the reader. The SOF- and EOF-detection also have to be done in software.

6.2 Hardware Requirements

The block diagram in figure 5 shows the main components of an RFID tag. The analog frontend is responsible for modulation and demodulation of data and for the power supply of the tag. The controller can be implemented in lots of different ways like as a microprocessor or a hardwired logic. It is responsible for implementing software requirements like data coding, implementation of the protocol commands, anti-collision mechanisms, and error detection. Depending on the used controller mechanism, parts of these requirements can be done in software or in hardware. The EPROM stores tag-specific data like the UID and the key for cryptographic algorithms. For implementation of the authentication protocol additional cryptographic hardware is needed that depends on the implemented algorithm. As the

die-size is very limited in RFID tags the architectures should be chosen well.

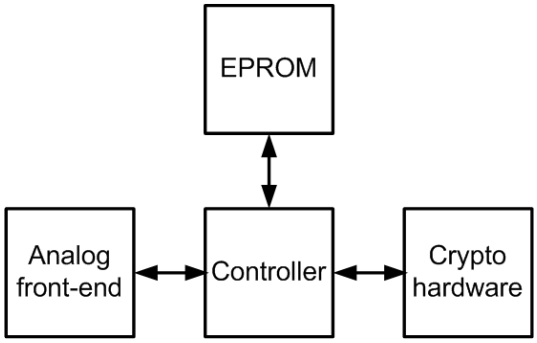

Figure 5: Block diagram of an RFID tag with authentication mechanism.

7 Results

The complete ISO/IEC 18000 standard protocol with the authentication extension was evaluated using Java models in different abstraction levels. The used encryption method for the authentication was implemented using the *IAIK Java Cryptography Extension (IAIK-JCE)* [6]. The highest abstraction level is just for algorithmic implementation of encryption and decryption. More details are shown in the model where reader and transponder are implemented as threads using sockets for simulation of the communication channel. Framing, collision detection, and CRC calculation are implemented to get test data for the hardware implementation.

Hardware is implemented for a fast prototype on an FPGA in VHDL. For controlling purpose a 4-bit RISC microcontroller is used with very limited resources. The microcontroller is an accumulator machine with a two-stage pipeline in Harvard architecture with a minimalist instruction set of 27 instructions. Memory-mapped IO is used for communication with the analog frontend module, the EPROM module, and the cryptographic hardware which is excluded in the current hardware implementation. The big advantage of using a microcontroller is that the cryptographic hardware could also be controlled using this microcontroller. This reuse reduces hardware costs.

The synthesis for a Xilinx FPGA Spartan-II XC2S200 results in 126 LUTs (Look-Up Tables) and a frequency of 35 MHz.

The synthesis for standard cells in a 0.35 μm CMOS process technology results in a chip area of 0.13 mm^2 for the 4-bit microcontroller excluding the program ROM. The estimation of the chip area of the entire system including microcontroller, program ROM, cryptographic hardware, and the analog frontend is about 1.5 mm^2.

8 Conclusions

The presented authentication protocol and its implementation is a novel solution for integrating security aspects on RFID tags. Since cryptography is not implemented on RFID tags in these days this proposal for an authentication protocol is completely new. The mechanisms how RFID tags work is presented and the existing protocol in the ISO/IEC 18000 standard is shown. An appropriated authentication mechanism (challenge-response protocol) was selected where the cryptographic algorithm is an AES implementation. The proposal of an FPGA implementation is shown and the associated chip area and power consumption estimations are presented. Future work will consist of implementation and verification of the cryptographic hardware on an FPGA.

References

[1] ISO/IEC 9798-3. *Information Technology – Security Techniques – Entity authentication mechanisms – Part 3: Entity authentication using a public key algorithm*. ISO, 1993.

[2] M. Aydos, T. Yanik, and Ç.K. Koç. An High-Speed ECC-based Wireless Authentication Protocol on an ARM Microprocessor. In *The 16th Annual Computer Security Applications Conference*, pages 401–409. IEEE Computer Society Press, December 11-15 2000.

[3] Klaus Finkenzeller. *RFID-Handbuch*. Carl Hanser Verlag München, third edition, 2002.

[4] International Organization for Standardization. *ISO/IEC 18000-3. Information Technology AIDC Techniques - RFID for Item Management*, March 2003.

[5] Xilinx Inc. *Spartan-II FPGA Family, Product Specification*. http://www.xilinx.com/, May 2003.

[6] Institute for Applied Information Processing and Communications (IAIK). *IAIK JCE Toolkit*. http://jce.iaik.tugraz.at/.

[7] Markus Jakobsson and David Pointcheval. Mutual authentication for low-power mobile devices. *Lecture Notes in Computer Science*, 2339:178–195, 2002.

[8] H. X. Mel and Doris M. Baker. *Cryptography Decrypted*. Addison-Wesley, Reading, MA, USA, 2001.

[9] Alfred J. Menezes, Paul C. Van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. The CRC Press series on discrete mathematics and its applications. CRC Press, 2000 N.W. Corporate Blvd., Boca Raton, FL 33431-9868, USA, 1997.

[10] J. Myers. RFC 2222: Simple Authentication and Security Layer (SASL), October 1997. Status: proposed standard. Updated by RFC2444.

[11] National Institute of Standards and Technology (NIST). *Entity Authentication Using Public Key Cryptography*. FIPS PUB 196, 1997.

[12] National Institute of Standards and Technology (NIST). *Advanced Encryption Standard (AES)*. FIPS PUB 197, November 2001.

[13] Bart Preneel. Cryptographic Primitives for Information Authentication - State of the Art. *Lecture Notes in Computer Science*, 1528:49–104, 1998.

[14] Douglas R. Stinson. *Cryptography - Theory and Practice*. CRC Press, second edition, February 2002.