Strong Authentication for RFID Systems Using the AES Algorithm[*]

Martin Feldhofer, Sandra Dominikus, and Johannes Wolkerstorfer

Institute for Applied Information Processing and Communications,
Graz University of Technology, Inffeldgasse 16a, 8010 Graz, Austria
{Martin.Feldhofer, Sandra.Dominikus,
Johannes.Wolkerstorfer}@iaik.tugraz.at

Abstract. Radio frequency identification (RFID) is an emerging technology which brings enormous productivity benefits in applications where objects have to be identified automatically. This paper presents issues concerning security and privacy of RFID systems which are heavily discussed in public. In contrast to the RFID community, which claims that cryptographic components are too costly for RFID tags, we describe a solution using strong symmetric authentication which is suitable for today's requirements regarding low power consumption and low die-size. We introduce an authentication protocol which serves as a proof of concept for authenticating an RFID tag to a reader device using the Advanced Encryption Standard (AES) as cryptographic primitive. The main part of this work is a novel approach of an AES hardware implementation which encrypts a 128-bit block of data within 1000 clock cycles and has a power consumption below 9 μA on a 0.35 μm CMOS process.

Keywords: Radio frequency identification (RFID), symmetric challenge-response, Advanced Encryption Standard (AES), low-power design.

1 Introduction

Radio frequency identification (RFID) systems are used for the automatic retrieval of data about goods, persons, or animals, more generally speaking: an object. The object is equipped with a small circuit, called RFID tag, and the information stored on the medium can be automatically retrieved by a reader device. This property can be used in industrial applications for tracking of goods or in access systems. RFID systems do not require line-of-sight and work contactless. Data and energy are transmitted via radio frequency.

Each RFID system consists of a tag, which is attached to the object to identify, and a reader which is able to retrieve data from the tag. The reader may as well be able to write data to the tag's memory. Additionally, to implement an application on data received from the tags, a host is used (see figure 1). Host commands are converted into reader requests and broadcasted via radio

[*] This work origins from the Austrian Government funded project *ART* established under the embedded system program FIT-IT.

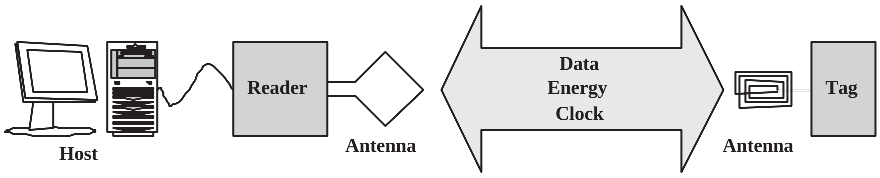

Fig. 1. Structure of an RFID system

frequency. If a tag is inside the reader's field, it sends a response. Tag responses can be processed by the host corresponding to the current application.

In this paper we concentrate on passive tags. This means that they receive their energy from the reader field. The field's intensity is limited by national and international regulations, so the power consumption of the tag as well underlies limitations. For this reason power-aware designing of the tag circuitry is necessary. Less power consumption also leads to longer reader ranges where tags can work with the available energy. More technical details about RFID systems (coupling mechanisms, data rates, data coding, modulation, frequency...) can be found in [4] and in RFID standards [7, 3].

RFID technology offers convincing benefits in many sectors. Industry and retailers save money by enhanced automation of fabrication and warehousing. Consumers can also take advantage from goods being able to communicate with their environment (e.g. washing machine communicates with clothes, milk packs communicate with refrigerator). Other applications for RFID systems are access control, animal tracking, proof of origin of goods, toll systems, car immobilization, and so on. There are even approaches to secure money with RFID tags [8]. So it looks like RFID will be a very popular technology in the near future.

Because of this popularity, people began to think about security and privacy issues concerning this technology. The public opinion was awakened by the "discovery" of consumer tracking: the ability of RFID readers to identify RFID tags in their environment could be used to track the movements of a consumer, who just bought an article equipped with an RFID tag. Another security concern is the forgery of tags, when RFID tags are used for access control, theft or loss control, or for proof of origin. The third security issue to be mentioned is the unauthorized access to tag's memory contents. If sensible data are stored in the memory, this is a serious security problem.

Enhanced security always comes along with extra costs. Although the industry claims low-cost tags, sooner or later the security issue has to be faced in order to make RFID an everyday technology. In this paper we propose the implementation of an authentication method for RFID systems using strong cryptography. The Advanced Encryption Standard (AES) is used as cryptographic primitive, because it is standardized and considered to be secure. We propose an authentication protocol and how it can be integrated into existing standards. Furthermore, we present a low-power implementation of the AES which is suitable for RFID tags in terms of power consumption and die size.

2 Security Considerations for RFID Systems

RFID systems are susceptible to security attacks: as they work non-line-of-sight and contactless, an attacker can work remote and passive attacks will not be noticed. Some of the main concerns are (unwanted) consumer tracking, tag forgery and the unauthorized access to the tag's memory content. These security risks have to be dealt with in order to gain a broad user acceptance.

2.1 Related Work

Some publications already deal with security aspects for RFID systems. Juels, Rivest, and Szydlo [9] propose so called "blocker tags" to protect consumers from tracking. One tag simulates a broad range of ID numbers, so a reader cannot identify it uniquely and tracking is circumvented. This was an approach to secure low-price tags which cost less than US$ 0.05.

Weis addresses the security and privacy issue of RFID systems in his master thesis [17] and in another publication together with Sarma, Rivest, and Engels [18]. He also deals with low-cost tags for industrial applications. He suggests a hash-lock mechanism as access control and a randomized version of it to deal with consumer tracking. He also presents some other concepts to enhance security in RFID tags. The assumptions made about the environment, for example that eavesdroppers cannot monitor signals from the tag, make the proposals sometimes not generally applicable.

Sarma, Weis, and Engels [15] also mention the problem of consumer tracking and recommend erasing the ID number of a tag at the point of sale as countermeasure. They also address the protection of tag contents and introduce the concept of access control through mutual authentication of tag and reader.

Juels and Pappu make a proposal how to secure banknotes with RFID tags [8]. They want to reach various security goals, some of them are consumer privacy, forgery resistance, and fraud detection. They propose a system which satisfies the requirements for all of the four main actors: the central bank, the merchants, the executive, and the consumer. The proposal in this publication is an RFID system using asymmetric cryptography combined with some methods which require physical access to the tag.

Finally, the "RFID Handbook" [4] deals with data security of RFID systems by using authentication mechanisms. This is also the topic of our paper. We propose using strong cryptographic algorithms to perform authentication.

2.2 Authentication

Authentication means that an object proves its claimed identity to its communication partner. This technique can solve all of the former mentioned security problems. Consumer tracking can be avoided, if tags only communicate their identity to authenticated readers. An unauthorized reader cannot get any information about tags which are currently in its field. The authentication of the reader to the tags also solves the unauthorized-access problem. Only authorized readers can read from or write to the tag's memory. Authentication of a tag means that the tag proves its identity to the reader. A forged tag cannot convince the reader of its authenticity, so forgery of tags can be circumvented.

Menezes, Oorschot, and Vanstone [12] differentiate between three authentication methods: password systems (weak authentication), challenge-response authentication (strong authentication), and customized and zero-knowledge authentication. Password systems offer a weak level of security and zero-knowledge techniques are often related to "strong" mathematical problems which are very costly in calculation and implementation. So we aim for the second type, the challenge-response techniques, which are broadly used.

There are asymmetric and symmetric challenge-response techniques. The disadvantage of asymmetric authentication methods is that they are very time consuming and costly to implement in hardware. So, they are not the first choice for RFID systems. There were attempts to design resource-saving asymmetric authentication algorithms. NTRU [5] has been proposed for RFID system implementations, but it was shown to have some security weaknesses [2, 11].

Symmetric methods work with one shared secret key. Authentication is done by proofing the possession of this secret key. The problem when using symmetric authentication methods is the key distribution and key management. Every update of the key has to be communicated to all participants. The compromising of only one device holding the key affects the whole system. This problems and some solutions were addressed in [12].

Symmetric authentication can be performed with encryption algorithms or can be based on keyed hash functions. Various reasons made the AES algorithm our favorite to use for the proposed authentication protocol. This encryption algorithm was chosen 2001 as encryption standard [13] and is considered to be highly secure. Furthermore, it is well suited for hardware implementations.

Protocols for symmetric challenge-response techniques based on encryption are defined in the ISO/IEC 9798-2 standard [6]. Unilateral authentication work as follows: there are two partners A and B. Both possess the same private key K. B sends a random number r_B to A. A encrypts the random number with the shared key K and sends it back to B. B proofs the result and can verify the identity (in other words the possession of K) of A.

$$A \leftarrow B : \qquad r_B \tag{1}$$

$$A \rightarrow B : \qquad E_K(r_B) \tag{2}$$

The mutual authentication protocol works similarly. B sends a random number to A. A encrypts r_B and a self-generated random number r_A with the shared key K and sends it to B. B decrypts the message and can proof if r_B is correct and gets r_A. B changes the sequence of the random numbers encrypts it with K and sends it to A. A proofs the result and verifies the identity of B.

$$A \leftarrow B : \qquad r_B \tag{3}$$

$$A \rightarrow B : \qquad E_K(r_A, r_B) \tag{4}$$

$$A \leftarrow B : \qquad E_K(r_B, r_A) \tag{5}$$

In order to minimize the power consumption and die size of the circuit, we decided to design our circuit for AES encryption only. By using modified protocols one-way authentication as well as mutual authentication can be performed, even if no AES decryption is available. Mutual authentication protocols require an additional random number generator, so they are more costly to implement.

2.3 Application Examples

Due to the key management problem, symmetric authentication methods are more suitable for closed systems. In closed systems each component can be controlled by one central instance. All devices can get their keys and key updates easily from the central control instance. In open systems, where the components can join the system unsolicited and no central control instance is available, key distribution and management is more difficult.

In airport luggage tracking systems each controlled bag can be equipped with an RFID tag and can be tracked throughout the whole airport. All assigned tag numbers are stored in a central server. In that way automated cargo and baggage transportation systems as well as security applications are possible (access control, checking if the holder of the bag is in the same plane as the luggage, etc.). Tag forgery and unauthorized access to the tag's memory should be avoided. Luggage tracking should also only be possible for authorized readers.

During transportation, RFID systems can be used to track and route goods on their way from the factory to the retailer. Here, theft and loss of goods can be rapidly detected, but the substitution with forged goods can only be avoided by using tag authentication mechanisms. Authentication of the reader could be used to prohibit unauthorized persons from spying the content of cargo. Tag memory can be used to record environmental incidents (for example, the disruption of the cold chain when transporting food). In this case, memory access must be secured by using reader authentication to prevent unauthorized memory manipulation.

Another application are car immobilizers, where symmetric authentication is already in use. In general, these implementations use proprietary encryption algorithms, which are optimized for the specific application. The security of these algorithms cannot be evaluated. Using AES would add some extra security. As a final example, for proof of origin of goods authentication is essential. In the next section we propose a method to integrate an one-way authentication protocol into existing RFID standards.

3 Security Protocol Design

In a security-enhanced RFID system, the level of security does not only rely upon the strength of the used cryptographic algorithms. The used protocols play a decisive role whether an attacker can successfully break into a system or not. Even if we use strong cryptographic algorithms, we need to ensure that the protocol is also secure. The protocol presented in this section allows the authentication of an RFID tag to a reader using the Advanced Encryption Standard (AES) [13]

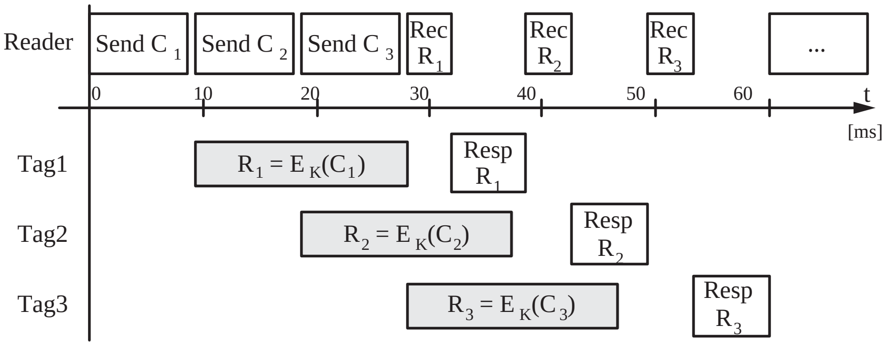

Fig. 2. Interleaved challenge-response protocol in RFID systems.

as the cryptographic primitive. In RFID systems, the limited computing power and low-power constraints of the tags require special considerations concerning the used protocols. In addition to the available bandwidth for data transmission, attention should be paid to the compatibility to existing standards like the ISO/IEC 18000 [7] or the Electronic Product Code (EPC) [3].

The protocol is based on the unilateral authentication mechanism using random numbers presented in section 2. Integrating the presented challenge-response authentication protocol into the ISO/IEC 18000 [7] standard requires some additional considerations. In addition to the mandatory commands, which all tags must implement, custom commands can be specified. The two commands, integrated for authentication, are sending a challenge to the tag and requesting the encrypted value. These commands extend the existing standard although the basic functionality remains unchanged. Due to the low-power restrictions, the internal clock frequency of the RFID tag must be divided from 13.56 MHz to 100 kHz. The applied standard demands that a response must follow 320 μs after a request. Otherwise, the tag has to stay quiet. This available time of 32 clock cycles at a frequency of 100 kHz is not enough for encrypting a challenge using the AES algorithm.

The solution to this problem is to modify the protocol as shown in figure 2. The challenges and the responses to the tags are interleaved to each other. Normally, there are a lot of RFID tags to be authenticated in the environment of a reader. After retrieving all unique IDs of the tags using the inventory request and the anti-collision sequence, the reader sends a challenge C_1 to *Tag1*. This tag immediately starts the encryption of the challenge without sending any response. In the meanwhile, the reader sends further challenges to the tags *Tag2* and *Tag3*. They also start encrypting their challenges after reception. After finishing the encryption of $E_K(C_1)$, *Tag1* waits for the request to send the encrypted value R_1 back to the reader. When the reader has sent the three challenges, it sends a request for receiving the response from $Tag1$. The received value R_1 is verified by encrypting the challenge C_1 and comparing the result with the received value. The two other unanswered challenges are received using the same

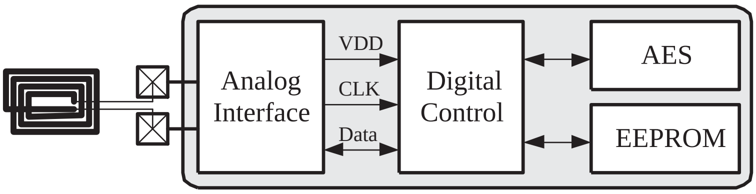

Fig. 3. Architecture of an RFID tag.

method. Then the reader starts from the beginning authenticating all other tags in the environment. This protocol was evaluated using high level models of the RFID communication channel and is a proof of concept for future research on authentication protocols in RFID systems.

This interleaving challenge-response protocol has the advantage that each tag has at least 18 ms (1800 clock cycles at a clock frequency of 100 kHz) time for encryption. A maximum of 50 tags can be authenticated per second. If there are only few tags in the range of a reader, the reader can decide to make breaks of at least 18 ms instead of sending interleaved requests.

4 RFID Tag Architecture

The architecture of a security-enhanced RFID tag is sketched in figure 3. It consists of four parts: analog frontend, digital controller, EEPROM, and AES module. The analog frontend is responsible for the power supply of the tag which is transmitted from the reader to the tag. Other tasks of the analog frontend are the modulation and demodulation of data and the clock recovery from the carrier frequency. The digital control unit is a finite state machine that handles communication with the reader, implements the anti-collision mechanism, and executes the commands in the protocol. Furthermore, it allows read and write access to the EEPROM and the AES module. The EEPROM stores tag-specific data like the unique ID and the cryptographic key. These data must be retained when the power supply is lost. The security-enhanced RFID tag calculates strong cryptographic authentication with an AES module which is designed for low-power requirements and low die-size restrictions. The requirements concerning power consumption and chip area and a description of the AES module are presented in the following sections.

4.1 Requirements for RFID Tag Design

In order to achieve a significant economic benefit from using RFID systems, tags will need to be priced under US$ 0.10 [15] for simple identification tags and a little bit higher for security-enhanced tags. Additionally to the aspect of low cost, the environmental conditions play a decisive role because contactless identification must work within a distance of a few meters. The limiting factors thereby are the available power supply for the tag and the signal strength for modulation and demodulation. The available power consumption for the digital part of the RFID tag (digital controller and AES module) is amounting to 20 μA.

Estimating the current consumption of the digital controller to 5 μA, 15 μA remain for the AES module which should not exceed a chip area of 5,000 gates. Additionally, the number of authenticated tags per second is about 50. As presented in chapter 3, this leads to an available time slot of 18 ms for encrypting a 128-bit block of data. Our proposed AES architecture, which is presented in section 4.2, encrypts in about 1000 clock cycles. As a consequence, the clock frequency of the AES module can be reduced under 100 kHz. This allows to reach the ambitious power consumption goal.

4.2 AES Architecture

The Advanced Encryption Standard (AES) is a symmetric encryption algorithm which was selected in 2001 by the National Institute of Standards and Technology (NIST) as the Federal Information Processing Standard FIPS-197 [13]. It operates on blocks of data, the so called State, that have a fixed size of 128 bits. The State is organized as a matrix of four rows and four columns of bytes. The defined key lengths are 128 bits, 192 bits, or 256 bits. Our implementation uses a fixed key size of 128 bits. As most symmetric ciphers, AES encrypts an input block by applying the same round function. The ten round function iterations alter the State by applying non-linear, linear, and key-dependent transformations. Each transforms the 128-bit State into a modified 128-bit State. Every byte of the State matrix is affected by these transformations:

1. **SubBytes** substitutes each byte of the State. This operation is non-linear. It is often implemented as a table look-up. Sometimes the SubBytes transformation is called S-Box operation.
2. **ShiftRows** rotates each row of the State by an offset. The actual value of the offset equals the row index, e.g. the first row is not rotated at all; the last row is rotated three bytes to the left.
3. **MixColumns** transforms columns of the State. It is a multiplication by a constant polynomial in an extension field of GF(2^8).
4. **AddRoundKey** combines the 128-bit State with a 128-bit round key by adding corresponding bits *mod* 2. This transformation corresponds to a XOR-operation of the State and the round key.

The calculation of the 128-bit round keys works by applying the KeySchedule function. The first round key is equal to the cipher key. The computation of all other round keys is based on the S-Box functionality and the Rcon operation.

AES is a flexible algorithm for hardware implementations. A large number of architectures are possible to cover the full range of applications. AES hardware implementations can be tailored for low die-size demands in embedded systems or can be optimized for high throughput in server applications. This flexibility of the AES algorithm was intended by its creators. They paid attention that the algorithm can be implemented on systems with different bus sizes. Efficient implementations are possible on 8-bit, 32-bit, 64-bit, and 128-bit platforms.

Although many AES hardware architectures have been proposed, none of the reported architectures meets the requirements of an AES module for RFID tags regarding low die-size and low power-consumption requirements. Nearly all of these architectures have GBit throughput rates as optimization goal. This is contrarious to our needs where throughput is not of concern. Only a few published AES architectures do not optimize throughput at any cost. To name some: [14, 1] are FPGA implementations and [10, 16] are ASIC implementations of AES which care about hardware efficiency. All these implementations do not unroll the AES rounds for sake of silicon size. The more S-Boxes are used, the less clock cycles are needed for encryption. The encryption-only AES processor of I. Verbauwhede et al. [16] is a 128-bit architecture that utilizes 32 S-Boxes. It is able to calculate one AES round in a single clock cycle. The compact 32-bit AES architecture of S. Mangard et al. in [10] is confident with four S-Boxes and takes eight cycles for one round. The FGPA implementations of N. Pramstaller et al. [14] and P. Chodowiec et al. [1] are 32-bit architectures too. They also use four S-Boxes. Four S-Boxes suit a 32-bit architecture as each S-Box substitutes 8 bits. The MixColumns operation and the ShiftRows operation are 32-bit operations too because they transform either four columns bytes or four row bytes of the AES State. The AddRoundKey operation (128-bit XOR) can also be split-up into 32-bit operations.

Implementing the AES algorithm as a 32-bit architecture allows to quarter the hardware resources compared to an 128-bit architecture [10, 1]. This comes at the expense of quadrupling the time for an AES encryption. The lower amount of hardware resources has a positive side effect on the power consumption: a quarter of hardware resources consumes only a quarter of power. This is an important feature for wireless devices where the average power consumption is an even more important quality aspect than the overall energy needed to encrypt one block. The overall energy consumption of a 32-bit architecture might be worse than for 128-bit architectures. But RFID tags offer neither the silicon space nor is the electromagnetic field strong enough to power an 128-bit datapath.

The power requirements for RFID tags are even too restrictive to allow the operation of a 32-bit AES implementation. Therefore, we decided to implement the AES algorithm as an 8-bit architecture instead of a 32-bit architecture. This novel approach for a hardware implementation of the AES algorithm is motivated by two reasons. First, an 8-bit architecture allows to decrease the number of S-Boxes from four to one to save silicon resources. Second, 8-bit operations consume significantly less power than 32-bit operations do. A penalty of an 8-bit architecture is the increased number of clock cycles for encryption. In RFID authentication applications an encryption lasting for 1000 cycles does not deteriorate the authentication throughput when several tags are authenticated concurrently (see section 3).

The architecture of the proposed 8-bit AES module is depicted in figure 4. It is presumably the smallest hardware implementation of the AES algorithm.

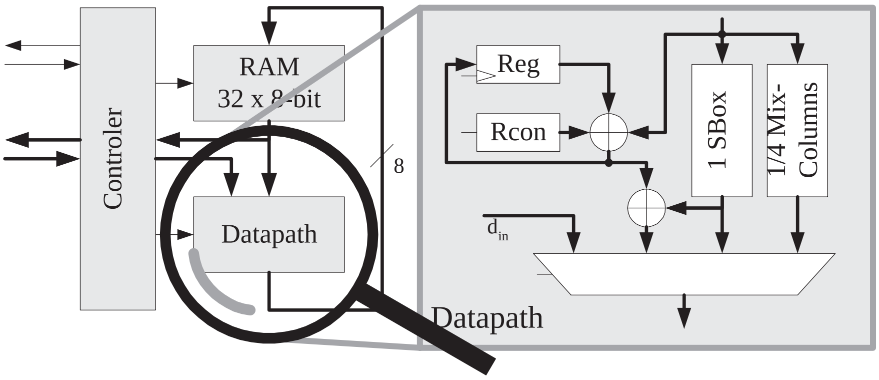

Fig. 4. Architecture of the AES module.

The module consists basically of three parts: a controller, RAM, and a datapath. The controller communicates with other modules on the tag to exchange data and it sequences the ten rounds of an AES encryption. Therefore, it addresses the RAM accordingly and generates control signals for the datapath. The RAM stores the 128-bit State and an 128-bit round key. These 256 bits are organized as 32 bytes to suit the intended 8-bit architecture. 32 bytes are the smallest possible memory configuration for AES. The memory is single ported to ease silicon implementation. Modified States and calculated round keys overwrite previous values. As no spare memory is present for storing intermediate values, the controller has to assure that no State byte nor a key byte is overwritten if it is needed again during encryption. The RAM implementation is register based. It makes use of clock gating to minimize power consumption. The datapath of the AES module contains combinational logic to calculate the AES transformations *SubBytes*, *MixColumns*, and *AddRoundKey* (see figure 4). The *ShiftRows* transformation is implemented by the controller. During the execution of SubBytes the controller addresses the RAM such that the ShiftRows operation is executed.

The biggest part of the AES datapath is the S-Box which is used for the SubBytes operation. There are several options for implementing an AES S-Box. The most obvious option is a 256 × 8-bit ROM to implement the 8-bit table lookup. Unfortunately, ROMs do not have good properties regarding low-power design. A more appropriate option is to calculate the substitution values using combinational logic as presented in [19]. We adapted the proposed combinational S-Box by omitting the decryption circuitry to suit our encryption-only AES. One feature of this S-Box is that it can be pipelined by inserting register stages. The S-Box makes use of one pipeline stage. This shortens the critical path of the S-Box to seven XOR gates and lowers glitching probability. Moreover, the pipeline register is used as intermediate storage for a pipelined SubBytes operation: during the substitution of one byte, the next byte is read from the memory. The substituted byte is written to the current read address. By choosing the read addresses properly this procedure combines efficiently the SubBytes and the ShiftRows operation. ShiftRows degrades to mere addressing.

Another innovative solution is the calculation of the MixColumns operation. We achieved to build a submodule which calculates only one fourth of the MixColumns operation. By accessing the submodule four times a complete MixColumns operation for one column of the State is executed. The MixColumns operation for one column is shown in equation 6. The equation reveals that all output bytes q_i of MixColumns are calculated by the same function—just the order of the input column bytes a_i differs.

$$
\begin{array}{c}
q(x) = a(x) \cdot c(x) = \bmod n(x) \qquad (6) \\
\hline
q_0 = (a_0 \otimes \{02\}) \oplus (a_3 \otimes \{01\}) \oplus (a_2 \otimes \{01\}) \oplus (a_1 \otimes \{03\}) \\
q_1 = (a_1 \otimes \{02\}) \oplus (a_0 \otimes \{01\}) \oplus (a_3 \otimes \{01\}) \oplus (a_2 \otimes \{03\}) \\
q_2 = (a_2 \otimes \{02\}) \oplus (a_1 \otimes \{01\}) \oplus (a_0 \otimes \{01\}) \oplus (a_3 \otimes \{03\}) \\
q_3 = (a_3 \otimes \{02\}) \oplus (a_2 \otimes \{01\}) \oplus (a_1 \otimes \{01\}) \oplus (a_0 \otimes \{03\}).
\end{array}
$$

We used this property to reduce the MixColumns circuitry to one fourth of its original size. The resulting circuit can calculate one output byte in one clock cycle. The $\frac{1}{4}$-*MixColumns* circuit contains, besides the combinational circuit to calculate q_i, three 8-bit registers to store three input column bytes a_i. These registers have to be filled before the first output q_i can be calculated. The fourth input a_i is taken directly from the RAM. Consequent output values are calculated by shifting the RAM output value to the registers and selecting the next value from RAM. The processing of one column takes seven clock cycles. A complete MixColumns operation to transform the entire State takes 28 clock cycles. The critical path of the MixColumns circuit is even shorter than the S-Box.

Remaining components of the datapath are the submodule Rcon, some XOR gates and an 8-bit register. Rcon is a simple circuit needed for key schedule. The XOR gates are needed for round key generation and are used to combine the State with the round key during the AddRoundKey transformation. The 8-bit register is needed during key schedule for storing intermediate results.

An encryption of a plaintext block works as follows. Before encryption is started the plaintext block has to be loaded into the RAM of the AES module. In the RFID tag application, the plaintext block is the 128-bit challenge which was received from the reader. The communication between the reader and tag is byte-oriented which fits nicely into the 8-bit architecture of the AES module: every received byte can be stored in the AES module. No intermediate memory is necessary. The cryptographic key is obtained in a similar way from the tag's EEPROM. Now the AES algorithm can be executed. It starts with a modification of the State by an AddRoundKey operation using the unaltered cipher key. Ten AES rounds follow by applying the transformations SubBytes, ShiftRows, MixColumns, and AddRoundKey. Only the last round lacks the MixColumns operation. Roundkeys are calculated just in time. This is usually called on-the-fly key schedule. The round key is derived from its predecessor by using the S-Box, the Rcon, and the XOR functionality of the datapath.

Table 1. Components and their complexity of the AES module.

Module/ Component	μA @100kHz	GE	Clock cycles
S-Box	0.67	395	280
MixColumns	0.41	252	288
AddRoundKey	0.53	90	144
KeySchedule	0.92	161	304
RAM	4.64	2,337	
Controller	0.98	360	
Total	**8.15**	**3,595**	**1,016**

Table 2. Comparison based on energy consumption, gate count, and clock cycles.

AES-128 Encryption	μA @100kHz	GE	Clock cycles
This work	8.15	3,628	992
Mangard [10]	47.24	10,799	64
Verbauwhede [16]	307	173K	10

5 Results

The implementation of the datapath of our AES-128 encryption design has a current consumption of 8.15 μA on a 0.35 μm CMOS process. It operates at a frequency of 100 kHz and needs 1,016 clock cycles for encryption of an 128-bit data block. The required hardware complexity is estimated to be 3,595 gate equivalents (GEs). All presented results come from simulations on transistor level. The complexity of each component is listed in table 1. Table 2 presents a comparison of our approach with the 32-bit implementation of S. Mangard et al. [10] and the 128-bit encryption-only AES processor of I. Verbauwhede et al. [16]. It can be seen that only our solution achieves the high demands for integrating cryptographic components into RFID tags. These requirements are the low power consumption and low die-size while conforming the requirements concerning encryption speed.

6 Conclusions and Future Research

This paper presented a security-enhanced RFID system which allows the strong cryptographic authentication. With this security-enhanced RFID systems, we pave the way for new security-demanding applications and for the everyday usage of RFID technology. A symmetric challenge-response authentication protocol was proposed which was integrated into the existing ISO/IEC 18000 standard. We showed an architecture for a low-power and low die-size implementation of the AES algorithm. The AES implementation has a chip area of 3,595 gates and has a current consumption of 8.15 μA at a frequency of 100 kHz. The encryption of 128 bits requires about 1000 clock cycles.

Future work will consist in the examination of advanced authentication protocols for one-way and mutual authentication. Other authentication methods (e.g. asymmetric techniques) should be analyzed for the suitability for RFID systems and circuits can be found for this purpose. In this way, the application range for RFID systems will be pushed further.

References

1. P. Chodowiec and K. Gaj. Very Compact FPGA Implementation of the AES Algorithm. In C. D. Walter, Çetin Kaya Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, September 8-10, 2003, Proceedings*, volume 2779 of *Lecture Notes in Computer Science*, pages 319–333. Springer, 2003.
2. W. Diffie and M. Hellman. Cryptanalysis of the NTRU Signature Scheme (NSS). In *ASIACRYPT: International Conference on the Theory and Application of Cryptology*, volume 2248 of *Lecture Notes in Computer Science*. Springer, June 2001.
3. EPCglobal. 13.56 MHz ISM Band Class 1 Radio Frequency (RF) Identification Tag Interface Specification. http://www.epcglobalinc.org/, February 2003.
4. K. Finkenzeller. *RFID-Handbook*. Carl Hanser Verlag München, 2nd edition, April 2003.
5. J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A Ring-Based Public Key Cryptosystem. In *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer, 1998.
6. International Organization for Standardization. *ISO/IEC 9798-2: Information Technology - Security techniques — Entity Authentication Mechanisms Part 2: Entity authentication using symmetric techniques*. ISO/IEC, 1993.
7. International Organization for Standardization. ISO/IEC 18000-3. Information Technology AIDC Techniques — RFID for Item Management, March 2003.
8. A. Juels and R. Pappu. Squealing Euros: Privacy protection in RFID-enabled banknotes. In *Financial Cryptography, 7th International Conference, FC 2003, Guadeloupe, French West Indies, January 27-30, 2003, Revised Papers*, volume 2742 of *Lecture Notes in Computer Science*, pages 103–121. Springer, 2003.
9. A. Juels, R. L. Rivest, and M. Szydlo. The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy. In *Proceedings of the 10th ACM Conference on Computer and Communication Security*, pages 103–111. ACM Press, 2003.
10. S. Mangard, M. Aigner, and S. Dominikus. A Highly Regular and Scalable AES Hardware Architecture. *IEEE Transactions on Computers*, 52(4):483–491, April 2003.
11. A. May. Cryptanalysis of NTRU. preprint, (unpublished), February 1999.
12. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997. Available online at http://www.cacr.math.uwaterloo.ca/hac/.
13. National Institute of Standards and Technology (NIST). FIPS-197: Advanced Encryption Standard, November 2001. Available online at http://www.itl.nist.gov/fipspubs/.
14. N. Pramstaller and J. Wolkerstorfer. An Efficient AES Implementation for Reconfigurable Devices. In *Austrochip 2003, Linz, Austria, October 1st, 2003, Proceedings*, pages 5–8, 2003.
15. S. E. Sarma, S. A. Weis, and D. W. Engels. RFID Systems and Security and Privacy Implications. In *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 of *Lecture Notes in Computer Science*, pages 454–470. Springer, 2002.
16. I. Verbauwhede, P. Schaumont, and H. Kuo. Design and Performance Testing of a 2.29 Gb/s Rijndael Processor. *IEEE Journal of Solid-State Circuits*, pages 569–572, March 2003.

17. S. A. Weis. Security and Privacy in Radio-Frequency Identification Devices. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA 02139, May 2003.
18. S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels. Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In *Security in Pervasive Computing, 1st Annual Conference on Security in Pervasive Computing, Boppard, Germany, March 12-14, 2003, Revised Papers*, volume 2802 of *Lecture Notes in Computer Science*, pages 201–212. Springer, 2004.
19. J. Wolkerstorfer, E. Oswald, and M. Lamberger. An ASIC implementation of the AES SBoxes. In *Topics in Cryptology - CT-RSA 2002, The Cryptographer's Track at the RSA Conference, 2002, San Jose, CA, USA, February 18-22, 2002*, volume 2271 of *Lecture Notes in Computer Science*, pages 67–78. Springer, 2002.