A Milestone Towards RFID Products Offering Asymmetric Authentication Based on Elliptic Curve Cryptography

Holger Bock[2], Michael Braun[1], Markus Dichtl[1], Erwin Hess[1], Johann Heyszl[2], Walter Kargl[2], Helmut Koroschetz[2], Bernd Meyer[1], and Hermann Seuschek[1]

[1] Siemens AG, Corporate Technology, Munich, Germany
[2] Infineon Technologies Austria AG, Graz, Austria

Abstract. After years of discussion within the RFID security community whether or not asymmetric cryptography would be feasible for RFID tags, we present a major breakthrough towards RFID products incorporating asymmetric authentication. For the challenge-response protocol applied, the response is calculated by performing an elliptic curve point multiplication using a random challenge and the tag's secret key. The design is resistant against side-channel attacks such as timing, simple power, differential power, and fault attacks. Some side-channel countermeasures depend on random numbers which are provided by a synthesizable true random number generator. The ISO/IEC 15693/18000-3 Mode 1 compliant RFID tag we present is based on a concept developed by Siemens Corporate Technology and published earlier this year. It incorporates Infineon's energy efficient 163 bit ECC engine with a single clock cycle addition and 41 clock cycles 4 bit parallel multiplication for the field operands. The energy spent during the ECC calculation is less than our target of $10\,\mu$J and the overall size of the tag's silicon is less than $0.8\,\mathrm{mm}^2$ in a 220 nm RFID technology. For early verification and host software development we designed an FPGA based emulation and test setup connected to an RF frontend. First samples of our ECC RFID chip are being manufactured and will be available in summer.

Keywords: RFID Security, Elliptic Curve Cryptography, Challenge-Response Protocol, Asymmetric Authentication

1 Introduction

A major challenge for RFID security is the integration of public-key cryptographic mechanisms into RFID tags. Such tags can be used in a wide variety of application scenarios, in which the authenticity of the RFID tags is required. For instance, authenticity of the tags is of concern if they are used to prevent counterfeiting of high-value products as pharmaceuticals or spare parts in avionics, where the quality of the products has a deep impact on people's safety, or to provide secure access control. Since most of these applications are decentralized, symmetric protocols have some major drawbacks, as the secret keys of the tags

must be stored inside the reader devices or a secure connection to a back-end database storing these keys must be provided. In order to prevent compromising such a system, the reader or the connection to the back-end database must be secured which is quite cost-intensive. Hence, public-key approaches are more reasonable in open-loop applications, since no secret keys must be handled by the reader. But the integration of public-key cryptography into such low-cost devices as RFID tags is a technological challenge. Tight constraints with respect to a small number of gate equivalents (GE) and low power consumption require highly optimized hardware implementations.

In this paper, we present the hardware implementation of an RFID tag with an integrated authentication module that uses public-key cryptography based on elliptic curves over binary fields. This work continues the investigations in [3] but now focuses on a detailed description from the architecture down to the circuit level. The result of the project is the hardware realization of a fully ISO/IEC 15693/18000-3 Mode 1 compliant tag with an asymmetric authentication unit.

2 Related Work

Recent publications, dealing with public-key cryptography on RFID tags, have all investigated the applicability of elliptic curves and proposed corresponding architectures.

Kumar and Paar [11] introduced a hardware implementation of elliptic curves over binary fields and provided a performance analysis with respect to chip size, memory, and computation time.

Batina et al. [2, 1] designed a similar elliptic curve arithmetic unit. They also provided a performance analysis and proposed a protocol for tag authentication based on Okamoto's identification protocol [15].

Fürbass and Wolkerstorfer [6] proposed a hardware design for an RFID authentication module which is based on elliptic curve digital signatures over prime fields.

A compact ECC processor including the estimate for the number of required gate equivalents was presented by Lee et al. [12].

A low-size hardware for elliptic curve scalar multiplication over binary fields can be found in Hein's master's thesis [8]. He implemented a complete circuit including the scalar multiplication control machines, local memory, and an interface for passive RFID tags.

A further approach based on elliptic curves defined over optimal normal bases can be found in [13].

3 The Authentication Protocol

In this section we describe the high-level tag authentication protocol which is a challenge-response procedure based on an elliptic curve Diffie-Hellman key exchange.

Let $y^2 + xy = x^3 + ax^2 + b$ be the equation of the elliptic curve $E = E(a, b)$ where $a, b \in \mathrm{GF}(2^n)$. Let $P = (x_P, y_P)$ be a point on the curve E and $k \in \mathbb{N}$ be a scalar with binary representation $k = (k_\ell, \ldots, k_1)_2$. The scalar multiplication $Q = k \cdot P = \sum_{i=1}^{k} P$ is done using Montgomery's method [14]. Given the affine x-coordinate x_P of P and the binary representation of the scalar k the algorithm computes a projective representation X_1/Z_1 of the x-coordinate of $k \cdot P$. If the result is the point at infinity then $Z_1 = 0$. In the following we use the term $(X_1, Z_1) \leftarrow \mathrm{Mul}(k, x_P)$ whenever Montgomery's algorithm is called.

Input: $k = (k_\ell, \ldots, k_1)_2$, x_P affine x-coordinate of P
Output: (X_1, Z_1) projective representation of the x-coordinate of $k \cdot P$
1. pick random value r
2. $X_1 \leftarrow r$, $Z_1 \leftarrow 0$, $X_2 \leftarrow rx_P$, $Z_2 \leftarrow r$
3 . for $i \leftarrow \ell$ downto 1 do
4. if $k_i = 1$ then
4.1. $T \leftarrow Z_1$, $Z_1 \leftarrow (X_1 Z_2 + X_2 Z_1)^2$,
4.2. $X_1 \leftarrow x_P Z_1 + X_1 X_2 T Z_2$, $T \leftarrow X_2$,
4.3. $X_2 \leftarrow X_2^4 + bZ_2^4$, $Z_2 \leftarrow T^2 Z_2^2$
5. else
5.1. $T \leftarrow Z_2$, $Z_2 \leftarrow (X_2 Z_1 + X_1 Z_2)^2$,
5.2. $X_2 \leftarrow x_P Z_2 + X_2 X_1 T Z_1$, $T \leftarrow X_1$,
5.3. $X_1 \leftarrow X_1^4 + bZ_1^4$, $Z_1 \leftarrow T^2 Z_1^2$
6. if $\Delta(X_1, Z_1, X_2, Z_2, x_P) \neq 0$ return error
7. else return (X_1, Z_1)

Fig. 1. Montgomery's method for scalar multiplication

Montgomery's algorithm depicted in Figure 1 also includes countermeasures against certain side-channel attacks. The algorithm itself is protected against simple power analysis since both branches of the if-else-statement execute the same sequence of operations. Hence the overall execution of the algorithm does not depend on the value of the secret scalar k. Differential power analysis is prevented by the randomization (steps 1 and 2) of the initial values. Finally, by evaluating the polynomial

$$\Delta(X_1, Z_1, X_2, Z_2, x_P) := x_3^2(X_1^2 Z_2^2 + Z_1^2 X_2^2) + X_1^2 X_2^2 + X_1 X_2 x_P Z_1 Z_2 + bZ_1^2 Z_2^2$$

we check whether faults have been induced during the execution of the for-loop (see [4]). A detailed investigation of side-channel attacks against our algorithm and the corresponding countermeasures can be found in [3].

In the remainder of this section we describe the authentication protocol. The protocol is based on a Diffie-Hellman key exchange and is shown in Figure 2.

Requirements. Let E be an elliptic curve over $\mathrm{GF}(2^n)$ and let $P = (x_P, y_P)$ be a point on E with prime order $q \in \mathbb{N}$. Let "GenSig" denote a public-key signature generation algorithm and let "VerifySig" be the corresponding signature

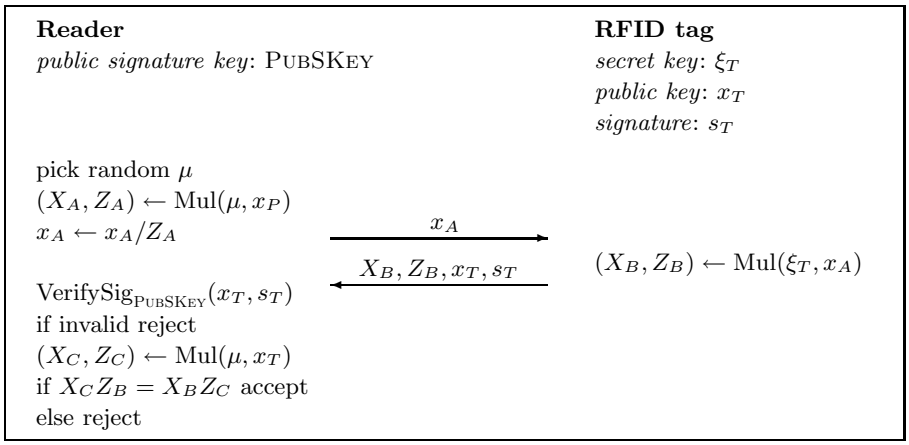

Fig. 2. Authentication protocol

verification algorithm. A key pair consisting of a private key for the signature generation and a public key for the signature verification is given by PRIVSKEY and PUBSKEY.

Setup Tag. Each tag is initialized with a randomly chosen private key $0 < \xi_T < q$ and a certificate (x_T, s_T) consisting of the corresponding public key x_T which is the affine x-coordinate of the point $T = \xi_T \cdot P$ and the signature s_T of x_T, generated with the signature generation algorithm "GenSig" using the private signature key PRIVSKEY, i.e. $s_T \leftarrow \text{GenSig}_{\text{PRIVSKEY}}(x_T)$.

Setup Reader. Each reader is initialized with the public signature key PUBSKEY. No further keys, in particular no secret keys, have to be stored on the reader.

Interaction. The reader randomly picks an ephemeral key $0 < \mu < q$, computes the affine x-coordinate x_A of the point $A = \mu \cdot P$ and sends x_A to the tag. Upon reception of this challenge the tag calculates $(X_B, Z_B) \leftarrow \text{Mul}(\xi_T, x_A)$. The result (X_B, Z_B) represents the x-coordinate X_B/Z_B of the point $B = \xi_T \cdot (\mu \cdot P)$. After computation the tag transmits the coordinates X_B, Z_B together with its certificate consisting of the public key x_T and the corresponding signature s_T back to the reader. Now the reader verifies the certificate by calling $\text{VerifySig}_{\text{PUBSKEY}}(x_T, s_T)$. If the signature is invalid the tag will not be accepted. Otherwise, the reader continues with the verification of the response. The reader calculates the projective coordinates X_C, Z_C of the point $C = \mu \cdot (\xi_T \cdot P)$ and checks if the x-coordinates X_C/Z_C and X_B/Z_B are equal. If the response is correct, the tag is accepted and declared to be authentic, otherwise the tag is rejected.

Protocol Security. The security of the protocol is based on the elliptic curve Diffie-Hellman problem. An attacker who wants to clone an authentic tag gets a challenge $A = \mu \cdot P$ from the reader and must return a valid response B together with an authentic public key $T = \xi_T \cdot P$ signed by the certification authority. Since the attacker only knows $A = \mu \cdot P$ and $T = \xi_T \cdot P$ he can determine the correct response $B = \xi_T \cdot (\mu \cdot P)$ if and only if he solves the corresponding Diffie-Hellman problem.

4 The Tag Architecture

The overall architecture of the tag is derived from a commercially available product, Infineon Technologies' SRF55V01P my-d™ light (see [10]) in order to optimize the development time and to minimize the risks for the functional prototype. Figure 3 shows the architecture of our ECC tag.

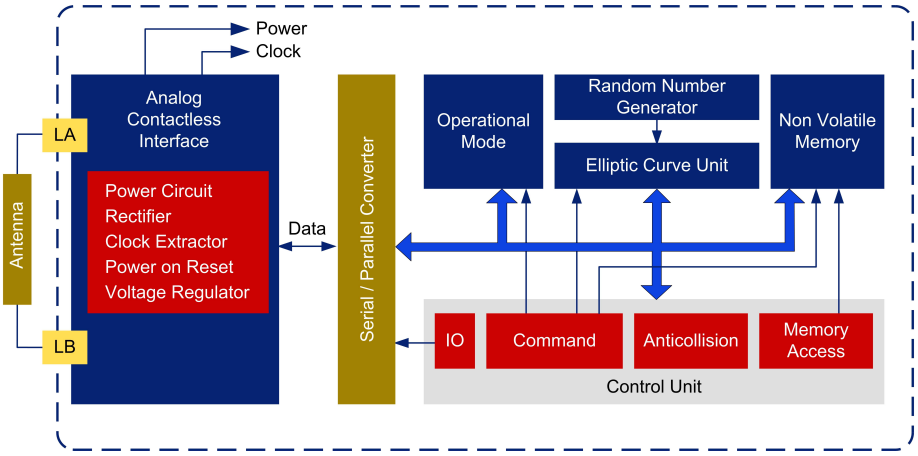

Fig. 3. Block diagram of the ECC tag

Analog Contactless Interface. The characteristics of the analog contactless interface can be optimized either for energy transmission and thus operating distance or for performance of the data transmission link. Final optimizations will be implemented when the prototype chip is available.

Control Unit. The digital part of the transceiver circuitry comprises a standard ISO/IEC 15693/18000-3 command decoder with the mandatory commands "Inventory" and "Stay quiet", and a variety of optional and custom commands.

In addition, the tag, being derived from the my-d™ light architecture, also supports the Infineon Technologies custom commands "Write Byte", "Destroy", "Check destroy", "Write pwd", and "Lock pwd".

Furthermore, we implemented the following new custom commands for the challenge-response interaction:

- "Get public data ECC", to make the tag send its public key together with a certificate,
- "Put challenge ECC", to send a 163 bit challenge from the reader to the tag,
- "Get response ECC", to retrieve the projective (X, Z)-coordinates of the response,
- "Write ECC", to personalize the tag, programming ECC data, i.e. the tag's key pair and the respective system certificate to the write-only area of the non-volatile memory,
- "Lock ECC", to write-protect the ECC data area.

Random Number Generator. The tag is provided with a very fast true random number generator based on a variant of the Galois Ring Oscillator (GARO) architecture. GAROs were introduced by J. Golić in [7], further results about them can be found in [5]. The original GARO design used a fixed feedback structure. It has turned out that a clock controlled version of the feedback structure provides a significant speed-up for the already very fast random number generation of GAROs. In order to minimize the hardware cost to implement this feature, only one position where the feedback is fed in depends on the clock. Of course, the two positions were chosen in such a way that the GARO does not have a fixed point for either value of the clock. Hence, the GARO cannot get stuck in a stationary state. The decision to use a GARO and not a FIRO (Fibonacci ring oscillator, also introduced in [7]) was based on the results of analog simulations of both random number generator architectures. The specific choice of the feedback configuration was also verified by analog simulation. The feedback of the GARO avoids very short feedback loops. The random number generator is used to prevent DPA (differential power analysis) of the computation of the Montgomery's algorithm; details are shown in Figure 1.

Non-Volatile Memory. The NVM is a byte oriented structured EEPROM based on a two-transistor cell architecture and offers 36 pages, four bytes each, resulting in a capacity of 144 bytes, which is twice the size of the NVM of the my-d™ light product. It serves as storage for

- 32 bytes of freely read- and writeable user data,
- 21 bytes for the tag's private key (163 bit) in a hardware protected write only area,
- 21 bytes for the tag's public key (163 bit),
- 50 bytes for the CA's signature of the tag's public key,
- 20 bytes of service data.

Figure 4 shows a layout of our RFID chip: The bottom left and right corners show the LA- / LB-pads to connect to the antenna, the lower third of the chip hosts the analog frontend including power supply and clock generation. On the

left hand side the NVM for storing the chip-individual private key and the corresponding certificate with the chip's public key and signature over the public key can be seen. The upper and the upper right part contain the synthesized logic for the ECC engine as well as the finite state machine processing the ISO/IEC 15693/18000-3 protocol and the logic of the NVM control circuits.

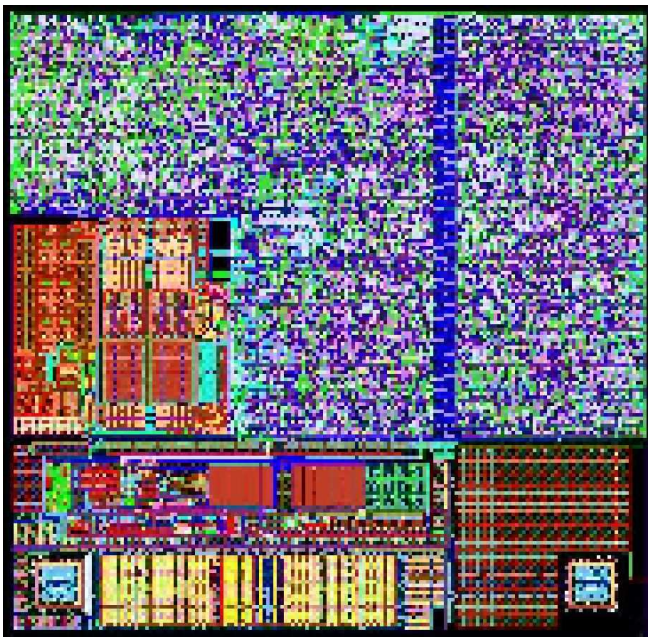

Fig. 4. Chip layout

5 The Elliptic Curve Engine

In this chapter we discuss the criteria leading to the specific implementation of the ECC engine.

5.1 Micro Architecture

The ECC hardware module consists of the following three building blocks:

- The *Arithmetic Unit* computes additions and multiplications in the finite field GF(2^n) and therefore contains an operand register and an accumulator register.
- The *Algorithm Control Unit* uses the arithmetic unit, the short-term storage unit, and the random number generator to implement the scalar multiplication as described in Figure 1. Random numbers are used to prevent DPA.

– The *Short-Term Storage Unit* stores up to five intermediate values of the scalar multiplication algorithm (Figure 1) and the challenge.

The secret scalar is stored in the RFID tag's non-volatile memory while the curve parameter and irreducible polynomial are hardwired (see Section 3).

5.2 Implementation Trade-Off

There are two major challenges for the implementation of the ECC hardware module. Since the number of gate equivalents directly corresponds to the cost of the chip, the primary goal is the minimization of this parameter. The second objective is the energy efficiency of our design. According to the baud rates and timings specified in ISO/IEC 15693/18000-3 the data transaction time of the public key and certificate amounts to approximately 50 ms. We decided that the ECC computation time has to be at most 100 ms. Based on existing RFID products we set 100 μW as maximum power consumption for the ECC module during computation. This amounts to a total maximum of 10 μJ energy available for one computation of the ECC module.

In order to reduce the total amount of gate equivalents we applied several techniques. Instead of flip-flops we used latches for the implementation of the short-term storage. The constant elliptic curve parameter and the irreducible polynomial, which defines the finite field, are hardwired. Furthermore, by analyzing the data-flow we optimized the number of required multiplexors. We also simplified the arithmetic unit by combining logic circuit elements for the multiplication and the addition. Finally, we reused the accumulator register in the arithmetic unit for shift-in of the challenge and the random numbers.

Some of these measures also result in a higher energy efficiency. Additionally, well-known methods in digital design have been applied to decrease the energy consumption while only slightly increasing the area. On the one hand we manually instantiated clock gates on register transfer level (RTL) and on the other hand clock gates were automatically inserted during synthesis. Furthermore, the outputs of the accumulator register were fixed because they drive large combinational nets inside the short-term storage unit.

After evaluating the energy consumption of the minimal architecture, measures to reduce energy consumption can be taken, if the requirements cannot be met. To do so, measures on the architectural level are most effective. We chose to investigate parallelizing the multiplication as e.g. proposed in [9] as architectural measure.

The multiplication in the binary field GF(2^n) is executed bit serial for minimal area requirements. Parallelizing the multiplication by 4 or 8 bits reduces the number of clock cycles used for the same computation. While the additional combinational logic consumes more energy, the overall energy consumption in the arithmetic unit is expected to be reduced significantly due to the decreased number of intermediate result updates in the sequential logic. The choice for the degrees of parallelization (4 and 8 bit) to explore was made according to

considerations concerning the interface to the ISO/IEC 15693/18000-3 protocol handling module and the non-volatile memory which is based on 8 bit blocks.

Four architectures have been implemented:

- The *1 bit architecture* includes latch based storage, hardwired constants, data path optimizations, functional and automatic clock gating, and fixing logic.
- The *4 bit architecture* further implements a 4 bit parallel multiplication in the arithmetic unit.
- The *8 bit architecture* realizes an 8 bit parallel multiplication in the arithmetic unit.
- The short-term storage unit contains 5 registers for 163 bit intermediate values to support our implementation of the scalar multiplication (Figure 1). However, by temporary use of the operand and accumulator registers one of these registers could be omitted. This concept demands increased complexity in the control unit. We expected this *4 bit without 5th register architecture* to significantly reduce area while consuming the same energy as the *4 bit architecture*.

Based on a clock frequency of 847 kHz which is 1/16 of the RF frequency as specified in ISO/IEC 15693/18000-3 we used PowerTheater[3] to obtain power consumption results for the synthesized standard cell net-list. These results and the gate counts after synthesis served as base for an analysis of the different architectures.

5.3 Trade-Off Results and Micro-Architecture Selection

Table 1 presents the gate counts of the four implemented architectures after synthesis.

Table 1. Synthesis results

Architecture	1 bit	4 bit	8 bit	4 bit w/o 5th reg
Gate Count	10392 GE	12876 GE	16247 GE	12536 GE
Difference to 1 bit		+ 2484 GE	+ 5855 GE	+ 2144 GE
Difference to 1 bit		+ 24 %	+ 56 %	+ 21 %
Difference to 4 bit			+ 3371 GE	− 340 GE
Difference to 4 bit			+ 26 %	− 3 %

In contrast to our expectations the area saved by the *4 bit without 5th register architecture* is not significant because of the increased complexity in the control

[3] `http://www.sequencedesign.com/solutions/powertheater.php`

state machine. The underlying source code also became hardly maintainable. Since the improvements in terms of area reduction are small, code complexity, however, had significantly increased, this version was discarded.

Table 2 lists the energy consumption as well as the computation time and mean power consumption of the three architectures of interest. According to these results the *1 bit architecture* is not eligible for implementation since it does not fulfill the energy requirement.

Table 2. Energy consumption, computation time, and mean power

Architecture	1 bit	4 bit	8 bit
Energy used for computation	15188 nJ	7511 nJ	7070 nJ
Difference to 1 bit		− 50 %	− 53 %
Difference to 4 bit	+ 102 %		− 6 %
Computation time at 847 kHz	330 ms	95 ms	56 ms
Mean power at 847 kHz	46 μW	79 μW	126 μW

The diagram in Figure 5 shows the energy over area of the three different architectures. It presents the trade-off between energy consumption and area. By investing additional 2150 GE to implement the 4 bit multiplication a 50 % reduction of energy consumption is achieved. We decided not to invest additional 3370 GE for the 8 bit multiplication to achieve only a 6 % decrease in energy consumption.

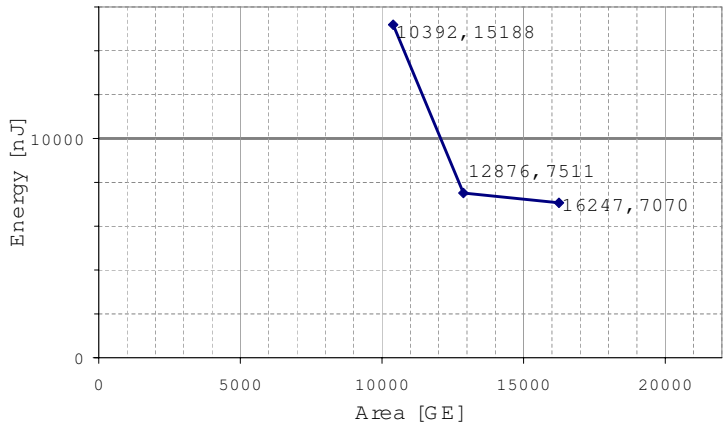

Fig. 5. Energy over required area

Thus, the *4 bit architecture* was chosen for physical implementation and prototype production. In our opinion it represents the best compromise with

respect to the power-energy trade-off. Table 3 presents the distribution of gate equivalents over the modules contained in the *4 bit architecture* ECC engine.

Table 3. Synthesis results – 4 bit architecture

Module	Gate Equivalents
ECC Hardware Module in Total	12876 GE
Algorithm Control State Machine	1432 GE
Arithmetic Unit	6171 GE
Storage	5273 GE

Including the ISO/IEC 15693/18000-3 protocol handling unit the digital part of the RFID tag requires 18k gate equivalents altogether. The implementation's standard cell layout uses 0.45 mm^2 of area at a utilization of 64 %.

After layout a power simulation of the digital circuit using extracted parasitic information generated results which are more precise regarding their absolute values. The simulated energy consumption of the ECC module is 9158 nJ which meets the target requirements. About 88 % of the energy is consumed in the arithmetic unit.

6 Verification and Test Setup

We employed a test and verification setup for early design verification and host software development. Figure 6 shows a block diagram which points out the functionality of our setup. The complete RFID digital design including the authentication module, the memory blocks, and the ISO/IEC 15693/18000-3 protocol engine was synthesized for an Altera[4] Cyclone EP1C20F324C7 FPGA. The FPGA is mounted on the prototyping board DIGILAB CC from El Camino[5]. In order to get full RFID functionality, the FPGA board is connected to a discrete RFID analog frontend.

The RFID frontend was designed for interfacing 13.56 MHz RFID readers. The modulation and demodulation is performed in hardware. Additionally, the frontend recovers the 13.56 MHz clock from the reader field and signals the presence of the field to the digital circuit. The frontend is powered by a 5 V supply voltage and all digital signals are 5 V CMOS compatible.

The I/O ports of the FPGA use 3.3 V levels. Therefore, level shifters must be inserted in the signal path. The conversion of the 5 V digital level to the 3.3 V level is done by a 3.3 V powered 74LVC244. For the opposite direction (3.3 V to 5 V) a 5 V powered 74HCT125 is used.

The RFID reader used for our setup is from Baltech[6]. This reader provides a special software API which allows the full control of the reader. Our test software

[4] `http://www.altera.com`
[5] `http://www.elcamino.de`
[6] `http://www.baltech.de`

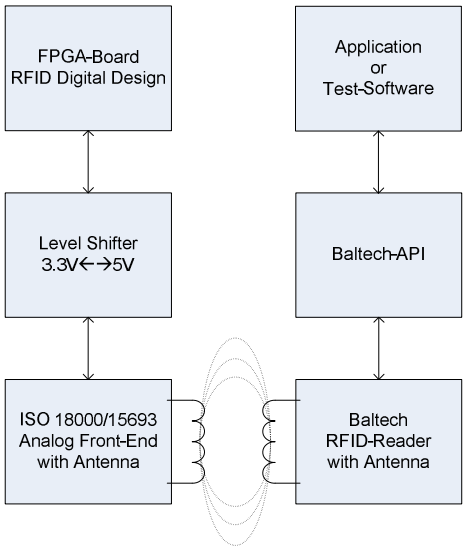

Fig. 6. Functional units and the signal flow of the RFID test assembly

uses raw commands to communicate with the tag directly. So the functionality of the RFID tag can be tested in a transparent manner.

Figure 7 shows the discrete assembly of the tag consisting of the FPGA board (left), the level shifters (center), and the analog frontend (right) with the characteristic loop antenna.

7 Outlook

Our chip is currently in the fab for manufacturing. We expect first demonstration samples after extensive functional and parametric verification by end of summer 2008. Afterwards, we will build up sample cards to support several demo applications, e.g. in the fields of brand protection and access control. Due to our FPGA based pre-silicon verification tool the software for the authentication protocol and all components for the tag and reader communications are already available.

Acknowledgement

First of all, we sincerely thank Anton Kargl for providing parameter sets of appropriate strong elliptic curves and for useful discussions on this work. In addition we appreciate very much the support we received from Albert Missoni for the analog contactless interface and during the tape-out phase and from Bernhard Roitner with respect to the verification setup. We are also grateful to Ulrike Meyer and Marcus Schafheutle for their valuable remarks to improve the editorial quality of this paper.

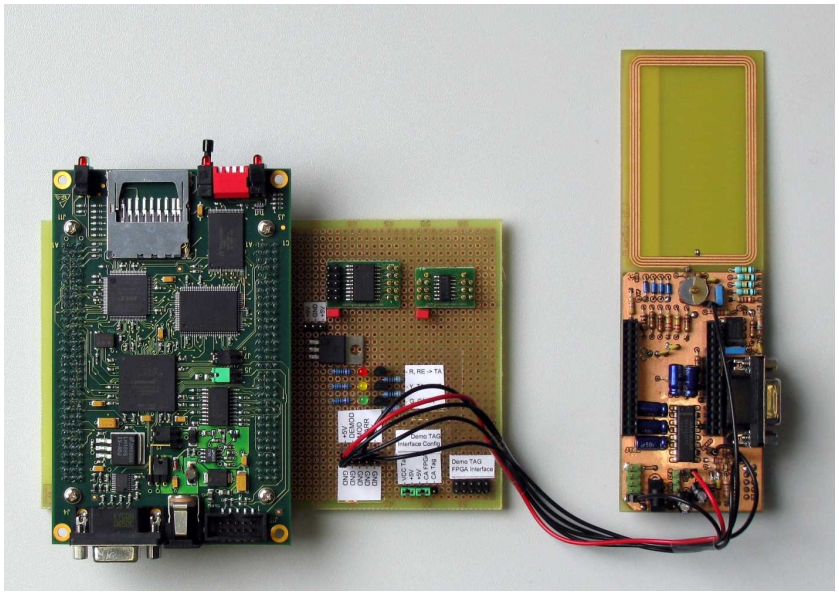

Fig. 7. Photograph of the discrete assembly of the prototype RFID tag

References

1. Lejla Batina, Jorge Guajardo, Tim Kerins, Nele Mentens, Pim Tuyls, and Ingrid Verbauwhede. Public-key cryptography for RFID-tags. Printed handout of Workshop on RFID Security — RFIDSec 06, July 2006.
2. Lejla Batina, Jorge Guajardo, Tim Kerins, Nele Mentens, Pim Tuyls, and Ingrid Verbauwhede. Public-key cryptography for RFID-tags. In *Fourth IEEE International Workshop on Pervasive Computing and Communication Security — PerSec 2007*, pages 217–222. IEEE, 2007.
3. Michael Braun, Erwin Hess, and Bernd Meyer. Using elliptic curves on RFID tags. *International Journal of Computer Science and Network Security*, 2:1–9, 2008.
4. Michael Braun, Anton Kargl, and Bernd Meyer. Efficient fault detection in Montgomery's method for scalar multiplication. To appear.
5. Markus Dichtl and Jovan Golić. High-speed true random number generation with logic gates only. In P. Paillier and I. Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems — CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 45–62. Springer-Verlag, 2007.
6. Franz Fürbass and Johannes Wolkerstorfer. ECC processor with small footprint for RFID applications. In *IEEE International Symposium on Circuits and Systems — ISCAS07*, 2007.
7. Jovan Golić. New methods for digital generation and postprocessing of random data. *IEEE Trans. Computers*, 55(10):1217–1229, October 2006.
8. Daniel Hein. Elliptic-curve cryptography suitable for RFID systems. Master's thesis, IAIK, Technical University of Graz, 2008. `http://www.iaik.tugraz.at/teaching/11_diplomarbeiten/archive/hein.htm`.

9. Markus Hütter. Generator aided full custom design of a datapath for operations in the galois field GF(2^n). Master's thesis, IAIK, Technical University of Graz, 2001.
10. Infineon. SRF 55V01P my-d light. Chip Card & Security ICs, 2005.
11. Sandeep Kumar and Christof Paar. Are standards compliant elliptic curve cryptosystems feasible on RFID? Printed handout of Workshop on RFID Security — RFIDSec 2006, July 2006. `http://events.iaik.tugraz.at/RFIDSec06`.
12. Yong Ki Lee, Kazuo Sakiyama, Lejla Batina, and Ingrid Verbauwhede. A compact ECC processor for pervasive computing. In *Secure Component and System Identification — SECSI 2008*, March 2008.
13. Pak-Keung Leung, Oliver Chiu-Sing Choy, Cheong-Fat Chan, and Kong-Pang Pun. An optimal normal basis elliptic curve cryptoprocessor for inductive RFID application. In *ISCAS*, 2006.
14. Julio Lopez and Ricardo Dahab. Fast multiplication on elliptic curves over GF(2^n) without precomputation. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 1999*, volume 1717 of *Lecture Notes in Computer Science*, pages 316–327. Springer-Verlag, 1999.
15. Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In E. F. Brickell, editor, *Advances in Cryptology — CRYPTO 1992*, volume 740 of *Lecture Notes in Computer Science*, pages 31–53. Springer-Verlag, 1992.