Small-footprint ALU for public-key processors for pervasive security *

Kazuo Sakiyama, Lejla Batina, Nele Mentens,
Bart Preneel and Ingrid Verbauwhede

Katholieke Universiteit Leuven, ESAT/COSIC, Kasteelpark Arenberg 10,
B-3001 Leuven-Heverlee, Belgium
{ksakiyam,lbatina,nmentens}@esat.kuleuven.be

Abstract. The field of embedded systems is constantly growing and new applications are emerging. Extreme examples are RFID tags and sensor nodes as they put new requirements on implementations of Public Key (PK) algorithms with a very low budget for the number of gates, power, bandwidth *etc.* This work describes a low-cost Modular Arithmetic Logic Unit (MALU) for Elliptic/Hyperelliptic Curve Cryptography (ECC/HECC) suitable for these applications. Our best solution for the MALU supporting ECC field arithmetic features 2 171 gates with an average power consumption of less than 40 μW in a 0.25 μm CMOS technology. This result is obtained by hardware resource sharing of the datapath and the usage of composite fields for ECC.

Key Words: pervasive computing, Elliptic/Hyperelliptic Curve Cryptography, composite fields, low-power hardware implementations

1 Introduction

The field of embedded systems is growing at a rapid rate, as devices such as mobile phones, PDAs, smart cards and key immobilizers are inescapable in our everyday life. Because these embedded devices are integrated into personal and professional infrastructures, the issue of security becomes paramount. In this case, protection of information is not enough: the embedded device itself can be lost or stolen and subject to various security attacks. An additional difficulty in making this wireless platform secure is that it is battery-operated and severely resource-constrained when compared to a server.

Hence, the distinguishing characteristics of embedded security can be divided into two categories: resource-limitation and physical accessibility.

* Kazuo Sakiyama, Lejla Batina and Nele Mentens are funded by FWO projects (G.0450.04, G.0141.03). This research has been also partially supported by the EU IST FP6 projects SCARD, SESOC, ECRYPT and IBBT.

The former one specifies severe resource constraints on the security architecture in terms of memory, computational capacity, and energy for embedded devices. Physical accessibility implies that a designer must assume that the device can and will fall into the hands of an adversary. Obvious examples are RFID tags and sensor nodes that can be easily accessed by everybody. This accessibility has led to a number of new security attacks in recent years [7].

The most challenging tasks for embedded security are implementations of Public-Key Cryptography (PKC). Public-key cryptosystems are present in almost all spheres of digital communication as they allow secure communications over insecure channels without prior exchange of a secret key. Other cryptographic services include authentication, key exchange and digital signatures. Although for example, authentication can be obtained by means of symmetric-key cryptography, it is evident that PKC substantially simplifies security protocols. In addition, the use of PKC reduces power due to less protocol overhead [4].

Two emerging examples of PKC applications are radio frequency identification tags (RFIDs) and sensor networks. They put new requirements on implementations of PK algorithms with very tight constraints in number of gates, power, bandwidth *etc.* In this work we show that the arithmetic unit for curve-based cryptosystems can be further optimized for these new challenging applications.

We investigate the possibility for PK services for pervasive computing. We show that ECC and HECC processors can be designed in such a way to qualify for lightweight applications suitable for RFID tags and wireless sensor networks. Here, the term lightweight assumes low die size and low power consumption. Therefore, we propose a hardware processor supporting ECC and HECC that features very low footprint and low-power. We investigate two types of solutions, one of which can be applied to ECC over binary fields $\mathbb{F}_{2^p}$ where p is a prime and another one to ECC over a composite field or for HECC on curves of genus 2. The latter implies the same arithmetic unit for both cases which is a factor 2 smaller than for the first ECC option.

The paper is organized as follows. In Sect. 2 we give an overview of related work. Section 3 gives some background information on curve-based cryptography and supporting arithmetic. In Sect. 4 we give details of our implementation of the Modular Arithmetic Logic Unit (MALU) for ECC/HECC. Our results are discussed in Sect. 5. Sect. 6 concludes the paper.

2 Related Work

Low-power and compact implementations became an important research area with the constant increase in the number of hand-held devices such as mobile phones, smart cards, PDAs *etc.* Goodman and Chandrakasan [6] proposed a low-power cryptographic processor for ECC over both types of finite fields. The power consumed in ultra-low-power mode (3 MHz at VDD = 0.7V) is at most 525 μW.

The work of Gaubatz *et al.* [5] discusses the necessity and the feasibility of PKC protocols in sensor networks. In [5], the authors investigated implementations of two algorithms for this purpose *i.e.* Rabin's scheme and NTRUEncrypt. The results for NTRUEncrypt are very appealing with 3 000 gates and power consumption of less than 20 μW at 500 kHz. In [4] the authors presented an architecture of an ECC processor which occupies an area of 18 720 gates and consumes less than 400 μW of power at 500 kHz. The field used was a prime field of order $\approx 2^{100}$.

The property of authentication for RFIDs can be achieved by symmetric as well as asymmetric primitives. Most of the previous work dealt with implementations of symmetric ciphers. The most notable example is the work of Feldhofer *et al.* [10], which considered the implementation of AES on an RFID tag. Recently, Wolkerstorfer presented a low-cost hardware design for ECC that is suitable for implementations of ECDSA on a small IC [17]. He also conjectured that ECC might be feasible on high-end RFID-tags.

3 Preliminaries

In this section we give some background information on ECC and HECC. We also discuss the strategy for low-power design.

3.1 ECC/HECC over binary fields

ECC relies on a group structure induced on an elliptic curve. A set of points on an elliptic curve together with the point at infinity, denoted ∞, and with point addition as binary operation has the structure of an abelian group. Here we consider finite fields of characteristic two. A non-supersingular elliptic curve E over $\mathbb{F}_{2^n}$ is defined as the set of solutions $(x, y) \in \mathbb{F}_{2^n} \times \mathbb{F}_{2^n}$ to the equation: $y^2 + xy = x^3 + ax^2 + b$ where $a, b \in \mathbb{F}_{2^n}, b \neq 0$, together with ∞.

HECC was proposed in 1988 by Koblitz [8] as a generalization of elliptic curve cryptography. In particular, elliptic curves can be viewed as a special case of hyperelliptic curves *i.e.* an EC is an HEC with genus $g = 1$. For a detailed mathematical background we refer to [1].

The main operation in any curve-based primitive is scalar multiplication which can be viewed as the top level operation. At the next (lower) level are the point/divisor group operations. The lowest level consists of finite field operations such as addition, subtraction, multiplication and inversion required to perform the group operations. The only difference between ECC and HECC is in the middle level that in this case consists of different sequences of operations. Those for HECC are a bit more complex when compared with the ECC point operation, but they use shorter operands.

In this work we deal with the case of elliptic curves over $\mathbb{F}_{2^p}$ with p prime (as recommended by standards) and also over composite fields $\mathbb{F}_{2^{2\cdot p}}$. In this case $\mathbb{F}_{2^{2\cdot p}}$ is a field of quadratic extension over $\mathbb{F}_{2^p}$, so we can write $\mathbb{F}_{2^{2\cdot p}} = \mathbb{F}_{2^p}[x]/(f(x))$, where $deg(f) = 2$. Here each element from the field $\mathbb{F}_{2^{2\cdot p}}$ is represented as $c = c_1 t + c_0$ where $c_0, c_1 \in \mathbb{F}_{2^p}$, and the multiplication in this field takes three multiplications in $\mathbb{F}_{2^p}$ plus four additions.

The clear advantage in using composite fields for ECC or HECC is the fact that the field arithmetic can be performed in a smaller field. This implies reduction of the arithmetic unit with a factor 2.

The Weil descent attack [3] was introduced and successfully targeted against EC over binary fields of composite degree n. Further research showed that some composite fields should be avoided for ECC [15, 12]. However, it was shown that the composite fields with degree $n = 2 \cdot p$ (*i.e.*, fields of quadratic extension over $\mathbb{F}_{2^p}$, where p is prime, remain secure against Weil Descent attacks and its variants [2].

3.2 Binary field arithmetic

From the formulae for point operations it is evident that we need to implement only multiplications and additions. Squaring is considered as a special case of multiplication in order to minimize the area and inversion is avoided by use of projective coordinates.

As mentioned above we also consider composite fields implementations. As typical examples we deal with the fields $\mathbb{F}_{2^{163}}$ and $\mathbb{F}_{2^{2\cdot 83}}$ to maintain almost the same level of security. In the latter the field $\mathbb{F}_{2^{2\cdot 83}}$ is represented as $\mathbb{F}_{2^{2\cdot 83}}[t] = \mathbb{F}_{2^{83}}/t^2 + t + 1$. In general, for composite fields

the field arithmetic translates to the arithmetic in the subfield. More precisely, one addition and one multiplication in $\mathbb{F}_{(2^p)^2}$ map to 2 additions and 3 multiplications + 4 additions in $\mathbb{F}_{2^p}$ respectively.

3.3 Low power applications

Here we discuss some strategies for lightweight implementations of public-key cryptography. In particular, we review the design principles and concrete measures to obtain a compact, low-power implementation of elliptic curve cryptosystems.

CMOS Power Consumption The power consumption of a CMOS gate can be written as:

$$P = P_{SW} + P_{SC} + P_{LK} \,,$$

where P_{SW}, P_{SC} and P_{LK} denote switching (or dynamic) power, short-circuit power and leakage (or stand-by) power respectively [11]. It is known that in older technologies (0.13 μm and above), the largest contribution comes from P_{SW}. On the other hand, in deep sub-micron processes, P_{LK} becomes critical. However, for the time being one can focus on switching power and assume that it influences the total power the most of all components. In more detail the power dissipation in a CMOS gate can be written as:

$$P = (0.5 C_L \cdot {V_{DD}}^2 + Q_{SC} \cdot V_{DD}) f_{Clock} E_{SW} + I_{LK} \cdot V_{DD} \,,$$

where V_{DD} is the supply voltage, Q_{SC} is the short-circuit charge, f_{Clock} is the operating frequency, C_L is the load capacitance and E_{SW} is the switching activity factor. The second term represents the static power dissipation due to the leakage current I_{LK}. The leakage current is directly determined by the number of gates and the process technology.

For low-power consumption as of today, it is necessary to minimize E_{SW}, the circuit size (for the current CMOS technology) and the operating frequency. This can be achieved among others by architectural decisions as well, which is one of the key-issues for lightweight applications. Therefore, one should distinguish between power consumption and energy efficiency with the latter one being even more crucial. More precisely, the metric that is typically used and that should be minimized accordingly is energy per processed bit E. This value can be calculated as $E = \frac{P}{throughput} \left[\frac{J}{bit}\right]$.

4 Modular Arithmetic Logic Unit (MALU)

In this section the architecture for the MALU is briefly explained. The datapath of the MALU is an MSB-first bit-serial $\mathbb{F}_{2^n}$ multiplier with digit size d as illustrated in Figure 4. This arithmetic unit computes $A(x)B(x)$ mod $P(x)$ where $A(x) = \sum a_i x^i$, $B(x) = \sum b_i x^i$ and $P(x) = \sum p_i x^i$. The proposed MALU computes $A(x)B(x)$ mod $P(x)$ by following the steps: The MALU$_n$ sums up three types of inputs which are $a_i B(x), m_i P(x)$ and $T(x)$, and then outputs the intermediate result, $T_{next}(x)$ by computing $T_{next}(x) = (T(x) + a_i B(x) + m_i P(x))x$ where $m_i = t_n$. By providing T_{next} as the next input T and repeating the same computation for n times, one can obtain the multiplication result.

Modular addition, $A(x) + C(x)$ mod $P(x)$ can be also supported on the same hardware logic by setting $C(x)$ to the register for $T(x)$ instead of resetting register $T(x)$ when initializing the MALU. This operation requires additional multiplexors and XORs, however the cost of this solution is cheaper compared to the case of having a separate modular adder. This type of hardware sharing is very important for such low-cost applications.

The proposed datapath is scalable in the digit size d which can be determined arbitrarily by exploring the best combination of performance and cost.

In Fig. 1 the architecture of our ALU is shown for finite fields operations in $\mathbb{F}_{2^{163}}$. To perform a finite field multiplication, the *cmd* value should be set to 1 and the operands should be loaded into registers A and B. The value stored in A is evaluated digit per digit from MSB to LSB. We denote the digit size by d. The result of the multiplication will be provided in register T after $\lceil \frac{163}{d} \rceil$ clock cycles. A finite field addition is performed by giving *cmd* the value 0, resetting register A and loading the operands into registers B and T. The value that is loaded into T is denoted by C in Fig. 1. After one clock cycle, the result of the addition is provided in register T. The *cmd* value makes sure only the last cell is used for this addition.

The cells inside the ALU all have the same structure, which is depicted in Fig. 2. A cell consists of a full-length array of AND-gates, a full-length array of XOR-gates and a smaller array of XOR-gates. The position of the XOR-gates in the latter array depends on the irreducible polynomial. In this case, the polynomial $P(x) = x^{163} + x^7 + x^6 + x^3 + 1$ is used. The *cmd* value determines whether the reduction needs to be done or not. In case of a finite field multiplication, the reduction is needed. For finite field addition , the reduction will not be performed.

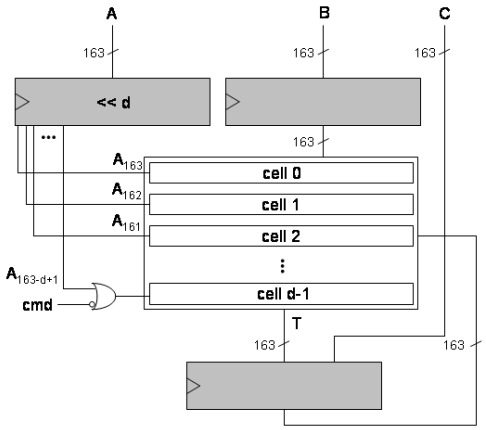

Fig. 1. Architecture of the MALU.

The output value T_{out} is either given (in a shifted way) to the next cell or to the output register T in Fig. 1. The input value T_{in} is either coming from the previous cell or from the output register T.

The strong part of this architecture is that it uses the same cell(s) for finite field multiplication and addition without a big overhead in multiplexors. This is achieved by using T as an output register as well as an input register. The flipflops in T are provided with a load input, which results in a smaller area overhead compared to a solution that would use a full-length array of multiplexors.

5 Results and Discussion

Now we give the results for the area complexity of both, ECC and HECC datapaths and the latency in the case of ECC. As mentioned above the core part of each curve-based protocol is one point/divisor multiplication. For example the protocol of Schnorr allows for authentication at the cost of only one point multiplication [14]. The results for various architectures with respect to the choice of fields and the size of d for ECC and HECC are given in Table 1 and in Table 2 respectively. For the case of ECC over composite fields and HECC the ALU shrinks in size but some speed-up

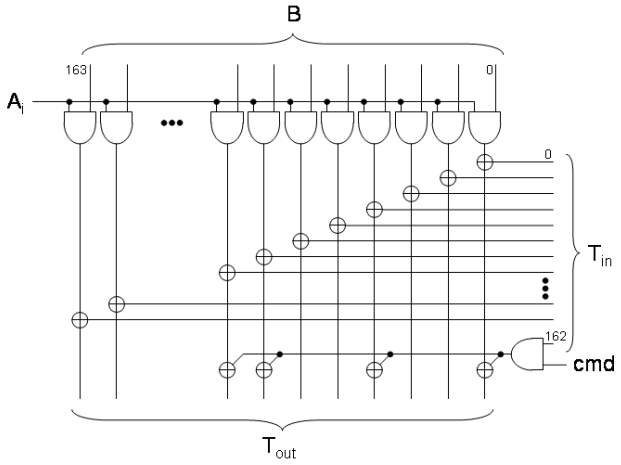

Fig. 2. Logic inside one cell of the MALU.

is then necessary which we obtain by means of digit-serial multiplications (instead of bit-serial one *i.e.* $d = 1$).

Table 1. The area complexity in gates of the MALU for various digit sizes for ECC.

Field size	d=1	d=2	d=3	d=4
131	4281	4758	5219	5685
139	4549	5043	5535	6028
151	4955	5472	6016	6540
163	5314	5900	6486	7052

Table 2. The area complexity in gates of the MALU for HECC and ECC over composite fields for various digit sizes.

Field size	d=1	d=2	d=4	d=6	d=8
67	2171	2421	2901	3420	3899
71	2299	2563	3069	3576	4083
79	2564	2854	3413	4012	4530
83	2693	2997	3582	4168	4794

The designs were synthesized by Synopsys Design Vision using a 0.25 μm CMOS library. One of our main goals for using composite fields was to reduce the ALU and the effect of that is visible in the table also. We notice that the ALU varies in size from 2 171 to 7 052 gates and the smallest one is obtained for the field $\mathbb{F}_{(2^{67})^2}$ and $d = 1$.

The graphical representations of our results are shown in Fig. 3 and Fig. 4. We can observe for ECC implementations that the upper bound for the number of gates of the ALU is slightly more than 7 kgates. An equivalent bound for HECC and/or ECC over composite fields is 4.8 kgates.

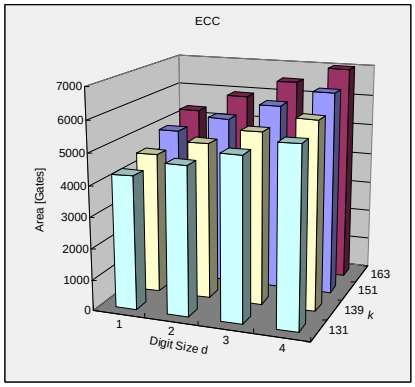

Fig. 3. Results for area complexity of the ECC-dedicated MALU for various bit and digit sizes.

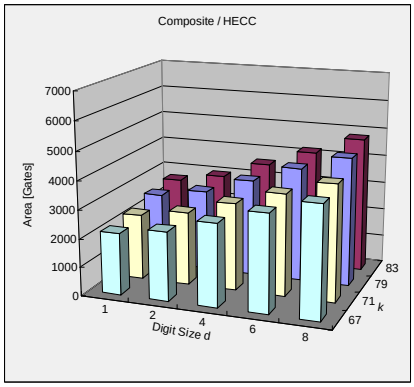

Fig. 4. Results for area complexity of MALU for composite fields and HECC for various bit and digit sizes.

We give some estimates for the performance. For the point multiplication one option is to use the method of Montgomery [13] that maintains the relationship $P_2 - P_1$ as invariant. In this case computations are performed on the x-coordinate only in affine coordinates (or on the X and Z coordinates in projective representation). That fact allows one to save registers which is one of the main criteria for obtaining a compact solution.

The performance in each case is calculated by the use of formulae for point operations as in [9] and we calculate the total number of cycles for each field operation by use of the following formulae for field operations. The total number of cycles for one field multiplication is $\lceil \frac{n}{d} \rceil + 3$ where n and d are the bit size of an arbitrary element from the field in which we are working and the digit size respectively. On the other hand, one

field addition takes 4 cycles. The number of cycles required for one point multiplication in the case of field $\mathbb{F}_{2^p}$, where p is a prime is: $n[13(\lceil \frac{n}{d} \rceil + 3) + 12]$. The results for the total number of cycles of one point multiplication for ECC over $\mathbb{F}_{2^{163}}$ and $\mathbb{F}_{(2^{83})^2}$ are given in Table 3.

Table 3. The number of cycles required for one point multiplication for ECC over the fields $\mathbb{F}_{2^{163}}$ and $\mathbb{F}_{(2^{83})^2}$.

Field size	d=1	d=2	d=3	d=4	d=6	d=8
163	353 710	182071	124 858	95 192	-	-
83	584 518	323 881	234 883	190 384	145 885	126 814

We do not include the timing for ECC over $\mathbb{F}_{2^{163}}$ when $d > 4$ because the area becomes too big. We observe that corresponding cases for two implementations *i.e.* those who result in similar performance are: $d = 1$ and $d = 2$, $d = 2$ and $d = 4$, $d = 3$ and $d = 8$. However, in each of those 3 cases the composite field implementation has a clear advantage as the ALU is much smaller.

The estimated power for the MALU is given in Fig. 5 and Fig. 6. To estimate the power consumption of the MALU, we used Design Vision from Synopsys. This power estimation is done with pre-layout netlists and typical wire-load models.

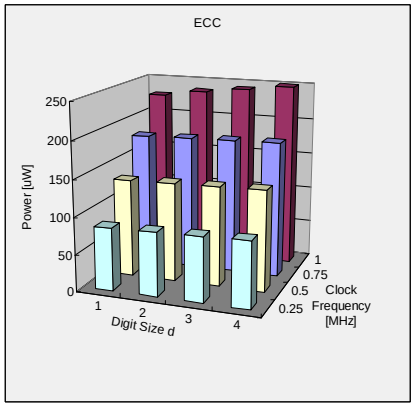

Fig. 5. Power consumed by ECC-dedicated MALU for various bit and digit sizes.

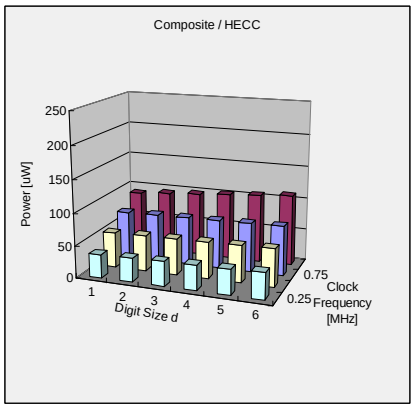

Fig. 6. Power consumed by MALU for composite fields and/or HECC for various bit and digit sizes.

To calculate the time for one point multiplications we need an operating frequency. However, the frequency that we can use is strictly influenced by the total power. As we did not have the total power measurements at the moment we assumed an operating frequency of 175 kHz as suggested in [16] in order to estimate the actual timing. We get 0.54 seconds for the best case of ECC over $\mathbb{F}_{2^{163}}$ ($d = 4$) and 0.72 seconds for the best case of ECC over composite fields ($d = 8$). We must add that some more storage is required for composite fields but this is not more than 20% according to our preliminary results.

6 Conclusions

This work gives a low-power and low footprint processor for ECC and HECC suitable for RFIDs and sensor nodes. We give detailed results for area and power and performance estimates for ECC over $\mathbb{F}_{2^p}$ with p prime and over composite fields $\mathbb{F}_{2^{2 \cdot p}}$. Furthermore, we propose to use the same ALU for HECC as for composite fields.

References

1. H. Cohen and G. Frey. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. Chapman & Hall/CRC, 2006.
2. C. Diem and J. Scholten. Cover Attacks - A report for the AREHCC project 2003. Technical report. `http://www.arehcc.com`.
3. G. Frey. How to disguise an elliptic curve (Weil descent). Presentation given at the 2nd Elliptic Curve Cryptography Workshop (ECC '98). Slides available at `http://www.cacr.math.uwaterloo.ca/`, September 14-16, 1998.
4. G. Gaubatz, J.-P. Kaps, E. Öztürk, and B. Sunar. State of the Art in Ultra-Low Power Public Key Cryptography for Wireless Sensor Networks. In *2nd IEEE International Workshop on Pervasive Computing and Communication Security (PerSec 2005)*, Kauai Island, Hawaii, March 2005.
5. G. Gaubatz, J.-P. Kaps, and B. Sunar. Public Key Cryptography in Sensor Networks - Revisited. In *1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004)*, Heidelberg, Germany, August 2004.
6. J. Goodman and A.P. Chandrakasan. An energy-efficient reconfigurable public-key cryptography processor. *IEEE Journal of Solid-State Circuits*, 36(11):1808–1820, November 2001.
7. A. Juels. RFID Security and Privacy: A Research Survey. *IEEE Journal on Selected Areas in Communications (IEEE J-SAC)*. to appear.
8. N. Koblitz. A family of Jacobians suitable for Discrete Log Cryptosystems. In S. Goldwasser, editor, *Advances in Cryptology: Proceedings of CRYPTO'88*, number 403 in Lecture Notes in Computer Science, pages 94–99. Springer-Verlag, 1988.
9. J. López and R. Dahab. Fast multiplication on elliptic curves over GF(2^m). In Ç. K. Koç and C. Paar, editors, *Proceedings of 1st International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 1717 of *Lecture Notes in Computer Science*, pages 316–327. Springer-Verlag, 1999.

10. J. Wolkerstorfer M. Feldhofer, S. Dominikus. Strong Authentication for RFID Systems using the AES Algorithm. In M. Joye and J. J. Quisquater, editors, *Proceedings of 6th International Workshop on Cryptographic Hardware in Embedded Systems (CHES)*, volume 3156 of *Lecture Notes in Computer Science*, pages 357–370. Springer-Verlag, 2004.
11. E. Macii. Design Techniques and Tools for Low-Power Digital Systems. course notes, 2003. IMEC course.
12. A. Menezes, E. Teske, and A. Weng. Weak fields for ECC. In Springer-Verlag, editor, *In Topics in Cryptology - CT-RSA - The Cryptographers' Track at the RSA Conference*, number 2964 in LNCS, pages 366–386, 2004.
13. P. Montgomery. Speeding the Pollard and Elliptic Curve Methods of Factorization. *Mathematics of Computation*, Vol. 48:243–264, 1987.
14. C.-P. Schnorr. Efficient Identification and Signatures for Smart Cards. In Gilles Brassard, editor, *Advances in Cryptology — CRYPTO '89*, volume LNCS 435, pages 239–252. Springer, 1989.
15. N.P. Smart. How secure are elliptic curves over composite extension fields? In B. Pfitzmann, editor, *Advances in Cryptology: Proceedings of EUROCRYPT'01*, number 2045 in Lecture Notes in Computer Science, pages 30–39. Springer-Verlag, 2001.
16. J. Wolkerstorfer. Is Elliptic-Curve Cryptography Suitable to Secure RFID Tags?, 2005. Workshop on RFID and Lightweight Crypto, Graz, Austria.
17. J. Wolkerstorfer. Scaling ECC Hardware to a Minimum. In ECRYPT workshop - Cryptographic Advances in Secure Hardware - CRASH 2005, September 6-7 2005. invited talk.