An Elliptic Curve Processor Suitable For RFID-Tags

L. Batina[1], J. Guajardo[2], T. Kerins[2], N. Mentens[1], P. Tuyls[2], and I. Verbauwhede[1]

[1] Katholieke Universiteit Leuven, ESAT/COSIC, Belgium
{Lejla.Batina,Nele.Mentens,Ingrid.Verbauwhede}@esat.kuleuven.be
[2] Philips Research Laboratories, Eindhoven, The Netherlands
{Jorge.Guajardo,Tim.Kerins,Pim.Tuyls}@philips.com

Abstract. RFID-Tags are small devices used for identification purposes in many applications nowadays. It is expected that they will enable many new applications and link the physical and the virtual world in the near future. Since the processing power of these devices is low, they are often in the line of fire when their security and privacy is concerned. It is widely believed that devices with such constrained resources can not carry out sufficient cryptographic operations to guarantee security in new applications. In this paper, we show that identification of RFID-Tags can reach high security levels. In particular, we show how secure identification protocols based on the DL problem on elliptic curves are implemented on a constrained device such as an RFID-Tag requiring between 8500 and 14000 gates, depending on the implementation characteristics. We investigate the case of elliptic curves over $\mathbb{F}_{2^p}$ with p prime and over composite fields $\mathbb{F}_{2^{2 \cdot p}}$. The implementations in this paper make RFID-Tags suitable for anti-counterfeiting purposes even in the off-line setting.

Key Words: RFID, counterfeiting, authentication, ECC, small area implementations

1 Introduction

RFID-tags are low-cost pervasive devices targeted at providing identification of goods. They consist of an antenna connected to a microchip. Because of the presence of this microchip, they can be considered as the next generation bar codes. One of their main advantages over bar codes is that they can be read out without line of sight. It is expected that in the near future trillions of these devices will be deployed. They will be used to identify goods and provide a link between the physical and the virtual world. It is predicted that this connection will lead to the next revolution after the Internet: The Internet of Things. Currently the main applications for RFID tags include: goods tracking in supply chain management, automated inventory management, automated quality control, access control, payment systems, *etc.* In the future, however, tagged items will also communicate with intelligent devices in the home (intelligent fridges, washing machines, *etc.*) and provide additional benefits to consumers. For example, a fridge will automatically detect whether the food is still OK and warn the consumer when necessary, the washing machine will detect the color of clothes in the washing and switch on the appropriate program, and, in general, home appliances will be *intelligent* and be able to communicate with other devices.

The fact that tags can be read without the need for line of sight, introduces a privacy threat. While walking home with tagged items in their bags, consumers can be scanned by unauthorized readers without their consent or permission. This potentially reveals privacy sensitive information about their preferences, things they buy, *etc.* New applications for RFID-Tags will introduce additional security risks. For instance, an emerging application that is being considered is the use of RFID-Tags for anti-counterfeiting purposes [25]. By locating an RFID-tag with specific product and reference information on a product, one can verify the authenticity of the product. This is done by running a secure protocol between a tag and a reader. If the required information is on the tag and verified to be authentic, the product is declared to be genuine and otherwise not. In a *cloning attack*, the attacker captures the necessary authentication information (obtained *e.g.*

by eavesdropping on the channel between the tag and the reader), and stores it in a new chip. In this way the attacker has effectively cloned the original tag. This clone cannot be distinguished from an original tag by a reader. In order to make the cloning of tags infeasible, it should not be possible to derive the tag secrets by active or passive attacks. Recently a lightweight version of such an authentication protocol was developed in [25]. The security of the protocol is based on the Learning Parity in the presence of Noise (LPN) problem. The protocol in [25] is proven secure against passive and against active adversaries in a detection-based model. Reference [49] suggests to use Schnorr's and Okamoto's identification protocols over Elliptic Curves (EC) to provide security against passive and active adversaries, respectively. In addition, [49] provides security against physical attacks as well, thanks to the physical properties of Physically Unclonable Functions (PUFs) [43, 14, 42]. The authors in [49] estimated that ECC and Hyper-Elliptic Cuve Cryptography (HECC) instances of secure identification protocols could be implemented requiring less than 5000 gates. However, memory requirements were not specified and an explicit construction is not provided.

The fact that tags have very constrained resources (memory, power, speed, area) but need security measures poses very interesting challenges to the security community. First, it is natural to investigate whether existing cryptographic algorithms can be implemented on a tag. Second, it encourages research for new protocols and algorithms targeted at resource constrained devices. Efficient implementations of AES for RFIDs have been investigated in [9], where it was shown that AES can be implemented in under 5000 gates. New lightweight protocols for RFID-Tags were developed in [25, 1]. To the authors' knowledge no implementations of ECC on RFID tags in less than 18,000 gates have been shown to be feasible. Moreover, the research community lacks consensus as to the feasibility of implementing public-key crypto-algorithms on (high-end) RFID tags. For example, [49] claim that public key cryptography on a tag is possible and [1] states: "Unfortunately asymmetric cryptography is too heavy to be implemented on a tag".

1.1 Our Contributions

We can summarize the contributions of this work as follows:

Feasibility of EC on RFID tags. We address the question of the implementation feasibility of EC based cryptography on a resource constrained device and, in particular, on an RFID-Tag. We answer this question *affirmatively*. We present ECC implementations of secure identification protocols such as Schnorr's [46] on an RFID-Tag. We show that by trading off performance for area it is possible to implement EC public-key cryptography on a tag. Since area is an important cost factor for the price of RFID-Tags, our main focus is to minimize the area required for the implementations. Our particular implementation of EC over binary fields has an area complexity of between 12k and 15k equivalent gates depending on the chosen field and implementation. This area complexity includes RAM and assumes a conservative estimate of 6 equivalent gates per RAM cell (i.e. a RAM cell instantiated as a flip-flop). If we were to use dedicated embedded RAM (see for example [40, 23]) our smallest design would require in the order of 8.2k equivalent gates. Notice that by todays standards this corresponds to a mid to high range tag. Although, it is anticipated that in the near future price pressure will continue to limit the number of gates in the ultra low cost tags, it can also be envisioned that eventually this number of gates will be available on all tags.

Trade off security for performance is acceptable. It is well known that smaller operand bit-lengths increase the computational efficiency of cryptographic operations (encryption, signature generation, etc.). However, this is not favored in the crypto community because of the reduced security offered by the resulting system. We, however, analyze the security that an

EC over a field $\mathbb{F}_{2^{131}}$ ($\mathbb{F}_{2^{142}}$) offers based on current state of the art attacks. We conclude that such fields offer acceptable security for many RFID applications including anti-counterfeiting. We based our definition of acceptable on the dollardays cost measure used in [28].

Our solution is based on identification schemes. We emphasize that our solution is based on identification schemes such as those of Schnorr or Okamoto. This is important because it provided us with an additional way to save area. In contrast, the solution in [52] is based on a challenge-response protocol (CRP) where an ECDSA signature needs to be computed. Such computation, requires the computation of a hash, thus requiring significant hardware resources in addition to the 23,000 equivalent gates of their smallest EC processor design. To our knowledge the best (area optimized) SHA-1 hardware implementation is that of [26], which requires about 4300 gates.

Apart from the contributions mentioned above, we would like to point out that by showing that ECC is feasible on RFID tags, we have indirectly solved another important problem. In particular, the availability of public-key cryptography for RFID-Tags makes the key management problem for a tag authentication system much easier. When authentication of tags is performed via symmetric-key cryptography, key distribution has to be carefully designed and inevitably it leads to several problems hindering the practicality or the security of the system. There are two extreme cases. One solution is to allow each tag to have a unique key that is shared with the backend system. This results in a very complex, expensive, and hard to manage system. On the other side of the spectrum, one has the situation where all tags share the same secret key. This implies that an attacker compromising a single tag, *e.g.* its own tag, also immediately compromises the whole system. This allows the attacker to clone any tag at will. Clearly, when public-key cryptography is used, the key management problem is solved. Additionally a single point of failure is easily avoided as well. The remainder of the paper is organized as follows. Section 2 gives a brief overview of related work. We present the protocols, EC algorithms and multiplier architectures used in our design in Sect. 3 and 4. In Sect. 5, we describe the processor architecture used for our prototype and estimate the size of the ALU, which is the part of the prototype that contributes the largest area to the overall design. In Sect. 6 we discuss and analyze the security provided by EC over non-standard field sizes. In addition, we argue that, based on the current state of the art, EC over $\mathbb{F}_{2^{131}}$ and over composite fields $\mathbb{F}_{2^{2p}}$, p a prime, offer adequate security for certain RFID applications. Finally, in Sect. 7 we discuss the results in detail and point out future work.

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.* Schroeppel *et al.* [47] presented a design for ECC over binary fields that was optimized for power, space and time in order to provide digital signatures. The processor in [47] had an area complexity of 191,000 gates. The work of Goodman and Chandrakasan [19] also dealt with energy-efficient solutions. They proposed a domain-specific reconfigurable cryptographic processor (DSRCP) for ECC over both types of finite fields. At 50 MHz, the processor operates at a supply voltage of 2 V and consumes at most 75 mW of power. In ultra-low-power mode (3 MHz at VDD = 0.7 V), the processor consumes at most 525 μW. Özturk *et al.* [41] introduced modulus scaling techniques that are applicable for ECC over a prime field to develop a low-power elliptic curve processor architecture. They obtained an ECC processor over the 166-bits long prime of size 30,333 gates with the performance of 31.9 msec for point multiplication.

The suitability of public-key (PK) algorithms for RFID is an open and important research problem as limitations in costs, area and power are quite severe. As already mentioned public-key cryptography (PKC) is a priori assumed to be impossible on RFIDs by many, which results in the lack of interest in the topic and a sparse previous work dealing with hardware implementations of PKC on low-power application platforms such as sensor nodes and RFID tags.

The work of Gaubatz *et al.* [16] discusses the necessity and the feasibility of PKC protocols in sensor networks. In [16], the authors investigated implementations of two algorithms for this purpose *i.e.* Rabin's scheme and NTRUEncrypt. The conclusion is that NTRUEncrypt features a suitable low-power and small footprint solution with a total complexity of 3000 gates and power consumption of less than 20 μW at 500 KHz. On the other hand, they showed that Rabin is not a feasible solution. In [15] the authors have compared the previous two algorithm implementations with an ECC solution for wireless sensor networks. The architecture of the ECC processor occupied 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 $\approx 2^{100}$.

RFID-based identification is an example of an emerging technology which requires authentication as a cryptographic service [10]. This property 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.* [31], which considered implementation of AES on an RFID tag. Recently, Wolkerstorfer [52] showed that ECC based PKC is feasible on RFID-tags by implementing the ECDSA on a small IC. This work is the first complete ECC low-power and compact implementation that meets the constraints imposed by the EPC standard. We compare the implementation of [52] with our results in Section 7 in more detail.

We consider here an ECC solution to provide identification for an RFID tag by means of Schnorr's identification scheme as discussed in [49]. If only resistance against passive attacks is needed, the Schnorr Identification scheme can be used as it is known to be secure against passive attacks under the discrete logarithm assumption. An alternative for providing more security is to use Okamoto's identification scheme [36], which is secure against passive, active and concurrent attacks under the DL assumption. However, we do not consider this protocol any further in this work.

3 Secure Identification Protocols

We investigate the protocol of Schnorr shown in Fig. 1. In this case a tag (prover) proves its identity to a reader (verifier) in a three-pass protocol. As it can be observed from the protocol,

1. **Common Input:** The set of system parameters in this case consists of: (q, a, b, P, n, h). Here, q specifies the finite field, a, b, define an elliptic curve, P is a point on the curve of order n and h is the cofactor. In the case of tag authentication, most of these parameters are assumed to be fixed.
2. **Prover-Tag Input:** The prover's secret a such that $Z = -a \cdot P$.
3. **Protocol:** The protocol involves exchange of the following messages:

Prover P		Verifier V
$r \in_R \mathbb{Z}_n$		
$X \leftarrow r \cdot P$	$\xrightarrow{X}$	
	$\xleftarrow{e}$	$e \in_R Z_{2^t}$
$y = ae + r \bmod n$	$\xrightarrow{y}$	
		If $y \cdot P + e \cdot Z = X$ then accept, else reject

Fig. 1. Schnorr's identification protocol.

the critical operation is the point multiplication. Thus, in the remainder of the paper, we describe a processor specifically suited for this operation and cheap enough that it is suitable for anti-counterfeiting RFID applications.

4 ECC implementations for RFID

In this section we elaborate on our choice of algorithms and we explain our strategy to minimize the area of the EC processor. Our strategy can be summarize as follows:

- We reduce the total number of intermediate registers for calculation of point operations.
- We use small digit sizes in our multiplier designs and investigate the effect of a dedicated squarer in the design's area and performance.
- We avoid having to recover the y-coordinate of the elliptic curve point in the tag and, in fact, only operate on the x-coordinate during the protocol. This is, in turn, helps us avoid having to compute two finite field inverses on the tag.

4.1 ECC Operations

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 ∞.

The point or scalar multiplication is the basic operation for cryptographic protocols and it is easily performed via repeated group operations. Here, we describe ECC operations at each level by following the top-down approach. For the point multiplication we chose Montgomery's [38], which maintains the relationship $P_2 - P_1$ as invariant. Montgomery's method is shown in Algorithm 1. It uses a representation where computations are performed on the x-coordinate only in affine coordinates (or on the X and Z coordinates in projective representation). That fact allows us to save registers which is one of the main criteria for obtaining a compact solution. We

Algorithm 1 Montgomery's Point Multiplication Method

Require: An integer k and a point $P \in E(\mathbb{F}_q)$
Ensure: $Q = k \cdot P$
Set $k \leftarrow (k_{n_k-1} \dots k_1 k_0)_2$
Set $P_1 \leftarrow P$, $P_2 \leftarrow 2 \cdot P_1$
for $i = n_k - 2$ downto 0 **do**
  if $k_i = 1$ **then**
    Set $P_1 \leftarrow P_1 + P_2$, $P_2 \leftarrow 2 \cdot P_2$
  else
    Set $P_2 \leftarrow P_2 + P_1$, $P_1 \leftarrow 2 \cdot P_1$
  end if
end for
$Q \leftarrow P1$
return Q

chose as starting point for our optimizations the formulae of Lopez and Dahab [30]. The original formulae in [30] require three intermediate registers (two for addition and one for doubling). In our case we eliminate two intermediate registers which added a few more steps to the original algorithms. The result of our optimizations are depicted in Algorithm 2. In this way we made a trade-off between performance and area as point operations require now 6 and 8 multiplications for point addition and doubling (instead of 5 and 6 M), respectively[3].

Algorithm 2 EC point addition and doubling: operations that minimize the number of registers

Require: $X_1, Z_1, X_2, Z_2, x_4 = x(P_2 - P_1)$	**Require:** $b \in \mathbb{F}_{2^n}$, X_1, Z_1
Ensure: $X(P_1 + P_2) = X(P_3) = X_3,\ Z_3$	**Ensure:** $X(2P_1) = X(P_5) = X_5,\ Z_5$
$Z_3 \leftarrow X_2 \cdot Z_1$	$Z_5 \leftarrow {Z_1}^2$
$X_3 \leftarrow X_1 \cdot Z_2$	$Z_5 \leftarrow {Z_5}^2$
$Z_3 \leftarrow X_3 + Z_3$	$Z_5 \leftarrow b \cdot Z_5$
$Z_3 \leftarrow {Z_3}^2$	$X_5 \leftarrow {X_1}^2$
$X_3 \leftarrow X_3 \cdot X_2$	$X_5 \leftarrow {X_5}^2$
$X_3 \leftarrow X_3 \cdot Z_1$	$X_5 \leftarrow X_5 + Z_5$
$T \leftarrow x_4 \cdot Z_3$	$Z_5 \leftarrow {X_1}^2$
$X_3 \leftarrow X_3 + T$	$Z_5 \leftarrow Z_5 \cdot Z_1$
	$Z_5 \leftarrow Z_5 \cdot Z_1$

4.2 $\mathbb{F}_{2^n}$ Arithmetic

Fields of characteristic two in polynomial basis were chosen for this investigation as arithmetic can be implemented efficiently and relatively cheaply in hardware over these fields. Although this is well understood, few previous attempts have been made to develop truly low area implementations of this arithmetic for ECC. Addition of two elements $c = a + b \in \mathbb{F}_{2^n}$ is performed via an n–bitwise logical XOR operation. The standard way to compute the product $c = a \cdot b \in \mathbb{F}_{2^n} \cong \mathbb{F}_2[x]/f(x)$, and $a = \sum_{i=0}^{n-1} a_i x^i$, $b = \sum_{j=0}^{n-1} b_j x^j$, $f = x^n + \sum_{i=0}^{s} f_i x^i$, $s < n$, is the one that uses convolution [3]

$$c = \sum_{j=0}^{n-1} \sum_{i=0}^{n-1} a_i b_j x^{i+j} \bmod f = a \sum_{j=0}^{n-1} b_j x^j \bmod f \tag{1}$$

This represents the most compact solution, where the $b_j a x^j$ partial products from (1) are computed iteratively and reduction modulo f of the degree n partial product polynomial is performed on each of the n iterations. The digit serial multiplication algorithm [27] may be considered as a generalization of this. Rather than processing the binary coefficients b_j of $b \in \mathbb{F}_{2^n}$ serially, a number of them are processed in parallel. Here there is scope to trade-off an increase in gate count for increased performance. This is an important consideration in low frequency implementations over relatively small (composite) fields as discussed here.

Here $b = \sum_{j=0}^{n-1} b_j x^j$, rather than being considered as n coefficients of $\mathbb{F}_2$ is considered as being composed of $d = \lceil \frac{n}{D} \rceil$ *words*, each word containing D elements of $\mathbb{F}_2$. Now $b = \sum_{k=0}^{d-1} \tilde{b}_k x^{kD}$, each $\tilde{b}_k = \sum_{l=0}^{D-1} b_{l+kD} x^l$, and

$$c = \sum_{k=0}^{d-1} (a\tilde{b_k}) x^{kD} \bmod f \tag{2}$$

can be calculated in d iterations. Notice that the $\tilde{b}_k a$ partial products are calculated recursively. A variant of the Song-Parhi method is illustrated as Algorithm 3. When $D = 1$ then $d = n$ and $\tilde{b}_k = b_j \in \mathbb{F}_2$ and this method reverts to Horner multiplication. Squaring $c = a^2 \in \mathbb{F}_{2^n}$ is a special case of multiplication [8]. It is well known that $a^2 = \sum_{i=0}^{n-1} a_i x^{2i}$ which can then be reduced modulo f to a field element in $\mathbb{F}_{2^n}$.

[3] Here, we count squarings also as multiplications.

Algorithm 3 Digit serial multiplication in $\mathbb{F}_{2^n}$

Require: $a = \sum_{i=0}^{n-1} a_i x^i$, $b = \sum_{k=0}^{d-1} \tilde{b_k} x^{kD}$ where $\tilde{b_k} = \sum_{j=0}^{D-1} b_{l_k} x^l$ and $f \in \mathbb{F}_2[x]$

Ensure: $c = a \cdot b \bmod f(x)$

```
c ← 0
for k from 1 to d − 1 do
    c ← x^D (c + b̃_{d−1} a) mod f
    b ← x^D b                                  {Only a D-bit left shift}
end for
c ← (c + b̃_{d−1} a) mod f
Return c
```

As mentioned in Sect. 6 for security reasons it is typically recommended to use fields $\mathbb{F}_{2^p}$ where p is a prime. As an example we investigate the cases where $p = 131$ and $p = 139$. However, we also consider EC over a quadratic extension of $\mathbb{F}_{2^p}$. For example, we consider $\mathbb{F}_{2^{134}} \equiv \mathbb{F}_{(2^{67})^2} \equiv \mathbb{F}_{2^{67}}[y]/g(y)$, where $\deg(g) = 2$ and g is an irreducible polynomial over $\mathbb{F}_{2^{67}}$. In this way we can translate the arithmetic from $\mathbb{F}_{(2^p)^2}$ to $\mathbb{F}_{2^p}$, which results in a reduction in the size of ALU by a factor of two approximately. In a composite field $\mathbb{F}_{(2^p)^2}$ each element can be represented as $z = xt + y$ where $x, y \in \mathbb{F}_{2^p}$.

4.3 Recovering the y coordinate of $Q = k \cdot P$

In traditional solutions, after computing $Q = k \cdot P$, one is required to transform back to affine coordinates and compute the y-coordinate of Q. We, however, advocate a different solution. One simple solution is to send both the end values of registers containing P_1 and P_2 in Algorithm 1 to the verifier so that the verifier himself can recover the y-coordinate of Q. This would incur in the sending of four finite field elements, corresponding to the projective coordinate representation of P_1 and P_2. Alternatively, the protocol can be run by only using the x-coordinates of all points involved. Notice that this is a rather old trick introduced by Miller in his seminal paper [37]. In either case, the projective coordinates sent to the verifier should be masked with a random value to avoid the attack described in [39]. This requires two extra multiplications at the end of the point multiplication which is negligible in comparison to the rest of the computation.

5 Elliptic Curve Processor Architecture

Our Elliptic Curve Processor (ECP) for RFID is shown in Fig. 2. The operational blocks are as follows: a Control Unit(CU), an Arithmetic Logic Unit (ALU), and Memory (RAM and ROM). In ROM the ECC parameters and the constants x_4 (the x-coordinate of $P_2 - P_1$) and b are stored. On the other hand, RAM contains all input and output variables and it therefore communicates with both, the ROM and the ALU.

The CU controls scalar multiplication and point operations. In the case of composite fields implementations, it also controls the operations in extension fields. In addition, the controller commands the ALU which performs field multiplication, addition and squaring. When the START signal is set, the bits of $k = \sum_{i=0}^{n_k - 1} k_i 2^i$, $k_i \in \{0, 1\}$, $n_k = \lceil \log_2 k \rceil$, are evaluated from MSB to LSB resulting in the assignment of new values for P_1 and P_2, dependent on the key-bit k_i. This is processed in an n-bit shift register. When all bits have been evaluated, an internal counter gives an END signal. The result of the last P1 calculation is written to the output register and the VALID output is set. The CU consists of a number of simple state machines and a counter and its area cost is small. The processor memory consists of the equivalent to seven n-bit ($n = p$) registers for ordinary fields and nine n-bit ($n = 2p$) registers for composite fields. Table 1 summarizes the

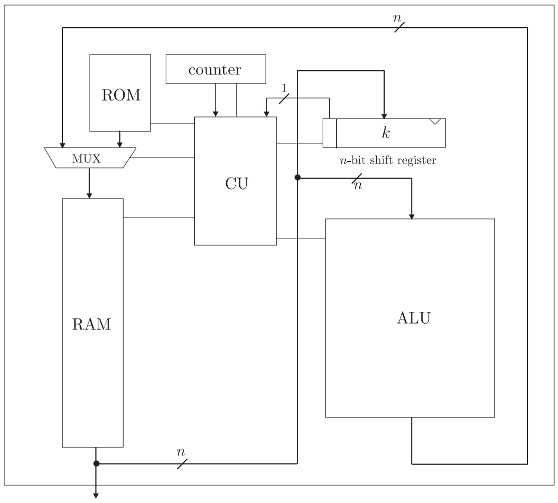

Fig. 2. ECP Architecture.

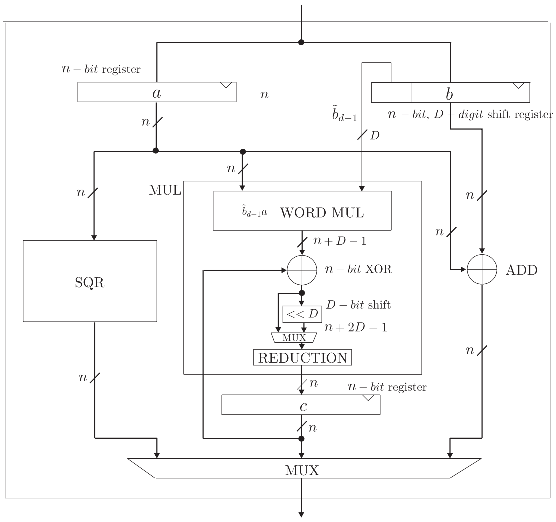

Fig. 3. ALU Architecture.

number of cycles required for basic operations and for a whole point multiplication in an EC over $\mathbb{F}_{2^p}$. The complexity for whole point multiplication over $\mathbb{F}_{2^{2p}}$ is , which can be obtained directly from Table 1 and Table 2.

Table 1. Cycle count for basic arithmetic operations and EC operations over $\mathbb{F}_{2^p}$. L: Load, C: Computation, S: Store

Operation	L	C	S	Total Cycles
$\mathbb{F}_{2^p}$ addition	2	1	1	4
$\mathbb{F}_{2^p}$ squaring	1	1	1	3
$\mathbb{F}_{2^p}$ multiplication	2	$\lceil \frac{p-1}{D} \rceil$	1	$\lceil \frac{p-1}{D} \rceil + 3$
EC operations assuming a squarer				
EC addition ($\mathbb{F}_{2^p}$)	$5\ MUL + 1\ SQ + 2\ ADD = 5\lceil \frac{p-1}{D} \rceil + 26$			
EC double ($\mathbb{F}_{2^p}$)	$3\ MUL + 5\ SQ + 1\ ADD = 3\lceil \frac{p-1}{D} \rceil + 27$			
Point mult. ($\mathbb{F}_{2^p}$)	$(n_k - 1)\left(8\lceil \frac{p-1}{D} \rceil + 53\right)$			
EC operations assuming no squarer				
EC addition ($\mathbb{F}_{2^p}$)	$6\ MUL + 2\ ADD = 6\lceil \frac{p-1}{D} \rceil + 26$			
EC double ($\mathbb{F}_{2^p}$)	$8\ MUL + 1\ ADD = 8\lceil \frac{p-1}{D} \rceil + 28$			
Point mult. ($\mathbb{F}_{2^p}$)	$(n_k - 1)\left(14\lceil \frac{p-1}{D} \rceil + 54\right)$			

5.1 The Arithmetic Logic Unit

The largest contribution in area to the overall design comes from the ALU, illustrated in Fig. 3. It consists of two n-bit registers a and c, and an n-bit shift-register b that outputs D bits at the time. In addition, the ALU has circuitry for implementing addition, squaring and multiplication in $\mathbb{F}_{2^n}$. Load and store operations between the ALU and memory cost a single clock cycle. The ADD block consists of n XOR gates, and the SQR block consists of at most $\frac{3n}{2}$ XOR gates (for particular irreducible polynomials this is known to be even cheaper [53]) and computes $\mathbb{F}_{2^n}$ additions and squarings in a single clock cycle once data has been loaded into the ALU. The MUL block implements an iteration of Step 3 of Algorithm 3 in a single clock cycle. Multiplication is calculated then in $d = \lceil \frac{n}{D} \rceil$ clock cycles. For composite fields, the field arithmetic translates to the arithmetic in the subfield as shown in Table 2.

Table 2. Basic operations in $\mathbb{F}_{(2^p)^2}$ expressed in terms of operations in $\mathbb{F}_{2^p}$.

Operation	Addition	Multiplication	Squaring
$\mathbb{F}_{2^p}$	2ADD	3MUL+4ADD	2SQR+ADD

5.2 ALU Complexity Estimates

The WORD MUL logic is performed in nD AND gates and nD XOR gates, and the intermediate addition operation operation costs another n XOR gates. Following the optimal polynomials in [27] we assume that $2D \leq m - s$. In this case the REDUCTION logic costs at most $5(2D-1)$ XOR gates for a pentanomial f. For a trinomial f the cost is at most $2D-1$ [53]. So the total hardware cost of the MUL in Figure 3. is at most $(D+1)n + 10D - 5$ XOR gates and nD AND gates. The total hardware cost of this ALU is thus at most, $3n$ 1-bit flip-flops, $n(D+2.5)+10D-5$ XOR gates, nD AND gates and $3n+2D-1$ 2:1 bit MUXes (for selecting the correct output). The control logic for counting the d multiplier operations and controlling the internal ALU operation is negligible in comparison.

5.3 A Word Regarding Power

At the present moment, we do not have an actual chip and we lack explicit power measurements for our simulations. Nevertheless, we believe that attaining the power values required for RFID applications using our design is possible. In fact, our processor architecture is very similar to the architecture presented in [52]. One particular characteristic of both designs is the usage of an Arithmetic Logic Unit (ALU) with a full-precision data path. The differences are on the details of our implementation: field size, choice of finite field arithmetic methodology (Montgomery vs. dedicated trinomial or pentanomial circuits), support for multiple fields versus support for a single field, hashing versus no hashing required in our implementation, etc. In general, our design is aimed at making our implementation as specific as possible to our particular application. This methodology leads to significant complexity reduction in the area requirements and thus also to power savings. Thus, since [52] was able to attain the power requirements of an RFID system, we are confident that our design, being smaller and simpler, will also attain the required power figures at the same or lower operating frequencies.

6 Security of EC over Non-Standard Finite Fields

Our EC implementation targets curves over 131-bit and 139-bit binary fields (both 131 and 139 are prime numbers). In addition, we have also considered composite fields of the form $\mathbb{F}_{2^{2 \cdot p}}$ where p is prime and the resulting fields are of order $\approx 2^{134}$ and 2^{142}, respectively. We decided to refer to these fields as "non-standard" as their bit-sizes are smaller than those *commonly* considered to be secure (i.e., $|\mathbb{F}_q| \geq 2^{160}$) and they include composite degrees, some of which are considered unsuitable for crypto applications due to the Weil Descent attack [11, 13, 18]. Our choice of fields is naturally related to the fact that smaller operand bit-lengths increase the computational efficiency of cryptographic operations while also providing adequate security for the applications in question. This practice is not favored in the crypto community because of the *reduced security* offered by the resulting system. Notice that we do not argue that the DL problem in an EC over $\mathbb{F}_{2^{131}}$ is easier to solve than over a $\mathbb{F}_{2^{163}}$ field, for example. However, in this section, we analyze the security of EC over non-standard fields based on current state of the art attacks and conclude that *such fields offer acceptable security* for many RFID applications including anti-counterfeiting. We based our definition of *acceptable* on the dollardays cost measure used in [28] and quantify it explicitly. We emphasize a point that has often been made in the past couple of years as we have seen a migration towards larger operand sizes: security is a risk assessment exercise. We believe that the risk of using EC-131 for RFID anti-counterfeiting applications is an acceptable one. We clarify what is meant by "an acceptable risk" in the next section.

6.1 Security of a 131-bit EC Implementation

For random elliptic curves $E(\mathbb{F}_q)$, the best known attack is Pollard's Rho algorithm [44], with complexity $\frac{\sqrt{\pi \cdot 2^{n-1}}}{2}$, where $q \approx 2^n$ and there is always a cofactor of at least 2 in the group order of $E(\mathbb{F}_q)$. Assuming this to be the best attack, [29] estimated that an EC defined over a 132-bit prime field provided a security level equivalent to a 952-bit RSA system (and a 70-bit block cipher) in the year 2000. In this context equivalent means that an attack against either RSA or EC requires approximately the same computational effort (measured in MIPS-years) to be successful. The year 2000 refers to the fact that in the model of [29], both EC over 132-bit field and 952-bit RSA provided the same security level in the year 2000 as DES did in 1982. We will write "DES(yyyy)" to mean "the security of DES in the year yyyy." On the other hand, the work in [28] takes a cost-based approach to evaluating the cost of an attack effort against a cryptosystem. This cost is measured in *dollardays* in [28]. The result is that according to [28], breaking 131-bit EC should be considered *cost* equivalent to breaking a 66-bit block cipher or 694-bit RSA. Thus, 131-bit EC would offer security equivalent to DES(1982) until the year 1996. In what follows, we will base our discussion on the cost-based approach of [28].

It might be tempting at this point to conclude that based on the previous paragraph a 131-bit EC does not provide adequate security for RFID applications. However, a closer look will indicate otherwise. First, notice that the notion of security in both [29, 28] is based on the assumption that DES provided adequate commercial security in the year 1982 and, in particular, that it could be cracked in one day with an investment of about US$40 million (40M dollardays). Clearly, not every key will be worth investing US$40 million to recover. In fact, only very large organizations or intelligence agencies would be able to invest such large sums of money to recover a key according to [4, 2]. To put the security of 131-bit EC into perspective, assume that your application requires security equivalent to DES(1993) or equivalently that using the model of [28] you would like security against an organization willing to invest about $\frac{100 \cdot 10^6}{2^{2 \cdot (1993-1980)/3}} \approx US\$300,000$ (a medium organization or illegitimate business according to [4]). Then, EC-131 would provide security equivalent to 858-bit RSA and equivalent to DES(1993) until 2007. In the previous discussion, we have not taken into account the fact that if certain hardware-based attacks are taken into account [50], it is recommended [2] to add 8 to 10 more bits to an EC defined over binary fields to obtain security equivalent to an EC defined over a prime field of similar size. However, as we will show in the next section, recent estimates show that EC over 131-bit binary fields provide security against medium and large organizations in the near to medium term, even assuming dedicated hardware attacks.

6.2 What Is Known Today?

In 2004, the ECC2-109 challenge was solved [7]. It was estimated that it required about $1.6 \cdot 10^{16}$ iterations and if running on a single dedicated Athlon XP 3200+ PC, it would have taken 1200 years. Based on this attack we estimate the cost of breaking the discrete logarithm (DL) problem on an EC over $\mathbb{F}_{2^{131}}$. First, note that the curve we used over $\mathbb{F}_{2^{131}}$ has order $\approx 2^{130}$. Hence, breaking the DL problem in $E(\mathbb{F}_{2^{131}})$ is by a factor of $\sqrt{2^{21}} \approx 1400$ times harder to break than the ECC2-109 challenge. Using the *dollardays* cost measure, the previous figures imply that the

ECC2-109 challenge cost was 1200 years · 365 days · 100 dollars ≈ 43 million dollardays[4] in 2004 and the 131-bit EC challenge would be in the billion of dollardays even today. Notice also that the 43 million dollardays figure implies that 131-bit EC offers comparable security to DES(1982). Thus, we feel that our selection of field order provides medium-term security which is sufficient for many applications intended for RFIDs including anti-counterfeiting.

Remark 1. Very recently there have been two proposals [21, 20, 5] for hardware machines intended for solving the DL problem in EC. In [20], the authors estimate that breaking the Certicom ECC-109 challenge (over prime fields) would take 30 days at a cost of US$3,000,000. Even if we were to assume that the same cost was required for breaking an EC implementation over $\mathbb{F}_{2^{131}}$, such an EC would require resources in the order of 90 million dollardays, thus comparable to DES(1982). The work in [5] proposes a design specifically tailored to binary fields. The authors estimate the time required to break the DL problem in an EC over $\mathbb{F}_{2^n}$ as $\frac{\sqrt{\frac{\pi \cdot 2^n}{2}} + 2^{n/3}}{100 \cdot 10^6}$ where $100 \cdot 10^6$ is the number of point additions per second that their FPGA-based processor can perform for an EC defined over $\mathbb{F}_{2^{79}}$. Then, assuming conservatively that the processor has the same throughput over $\mathbb{F}_{2^{131}}$, and an FPGA unit price[5] of US$50, the DL problem over $E(\mathbb{F}_{2^{131}})$ could be solved at a cost of $\frac{\sqrt{\frac{\pi \cdot 2^{130}}{2}} + 2^{130/3}}{100 \cdot 10^6 \cdot 3600 \cdot 24}$ days · 50 dollars ≈ 267 million dollardays, thus also comparable to DES(1982).

6.3 Security of EC over Composite Fields

In [11], the Weil descent attack is introduced against EC defined over binary fields of composite degree $n = k \cdot m$. At the time, it appeared that this work effectively rendered all composite field implementations of EC insecure. However, closer examination has demonstrated that composite fields with degree $n = 2 \cdot p$ (i.e., extension of degree two), where p is prime, remain secure against Weil Descent attacks and its variants. Notice that EC over composite fields $\mathbb{F}_{2^{2 \cdot p}}$, p a prime, have been previously proposed in the literature for efficient implementations [6, 47] and it was shown that the Weil descent attack is not applicable to them [6].

7 Results and discussion

In this section, we provide estimates for the latency and the area complexity of Schnorr's protocol. As mentioned above the core part of the protocol is one point multiplication. The results for various architectures are given in Tables 3 and 4. We considered solutions with or without the squarer as it allows also for a trade-off between area and performance. For the case of composite fields the ALU shrinks in size but some speed-up is then necessary which we obtain by means of a digit-serial multiplier (instead of a bit-serial one, *i.e.,* $D = 1$). The performance in each case is calculated by use of formulae for point operations as in Algorithm 2 and we calculate the total number of cycles for each case assuming the numbers for field arithmetic provided in Sect. 5. For ECC over composite fields we also use Table 2.

According to the systematic evaluation of Wolkerstorfer [51], ECC could meet low-power requirements on RFIDs assuming clock frequency of 175 kHz on 180 nm technology. The reason is that his area estimates exceeded the size that was expected to be acceptable for RFIDs (1 mm^2). However, we show that it is possible to implement ECC with less area at the cost of a decrease in functionality when compared to the one described in [51] and the use of smaller

[4] We have assumed that an Athlon XP 3200+ cost in 2003-2004 US$100 which is a conservative estimate based on [45].

[5] Reference [20] estimate the cost of a Xilinx XC3S1000 FPGA at US$50 for low quantities. The Virtex 4 used in [5] is a higher complexity FPGA, thus this price assumption should provide us with a conservative estimate.

field sizes. In addition, we chose the suggested frequency because it still results in a reasonable performance for RFID applications while reducing power consumption.

Table 3. Implementation results @ 175 kHz and assuming a dedicated squarer circuit.

Implementation		ALU	RAM	Perf. @175 kHz	Area wo RAM	AT factor	AT f.
Digit size	Field Type	[kgates]	[bits]	[s]	[kgates]	[wo. RAM]	[w. RAM]
D=1	$\mathbb{F}_{2^{131}}$	6306	917	0.81	8582	6975	11446
	$\mathbb{F}_{(2^{67})^2}$	3274	1206	1.44	6074	8734	19139
	$\mathbb{F}_{2^{139}}$	6690	973	0.91	9044	8259	13590
D=2	$\mathbb{F}_{2^{131}}$	6962	917	0.43	9233	3937	6284
	$\mathbb{F}_{(2^{67})^2}$	3610	1206	0.84	6410	5359	11409
	$\mathbb{F}_{2^{139}}$	7379	973	0.48	9734	4652	7442
	$\mathbb{F}_{(2^{71})^2}$	3648	1278	0.92	6534	6044	13137
D=3	$\mathbb{F}_{(2^{67})^2}$	3789	1206	0.64	6589	4187	8784
	$\mathbb{F}_{(2^{71})^2}$	3833	1278	0.71	6719	4786	10248
D=4	$\mathbb{F}_{(2^{67})^2}$	4103	1206	0.54	6903	3757	7694
	$\mathbb{F}_{(2^{71})^2}$	4152	1278	0.60	7038	4197	8769

Table 4. Implementation results @ 175 kHz and assuming no dedicated squarer circuit.

Implementation		ALU	RAM	Perf. @175 kHz	Area wo RAM	AT factor	AT f.
Digit size	Field Type	[kgates]	[bits]	[s]	[kgates]	[wo RAM]	[wRAM]
D=1	$\mathbb{F}_{2^{131}}$	5679	917	1.39	7953	11072	18731
	$\mathbb{F}_{(2^{67})^2}$	2953	1206	2.39	5708	13648	30949
	$\mathbb{F}_{2^{139}}$	6018	973	1.57	8380	13124	22267
D=2	$\mathbb{F}_{2^{131}}$	6335	917	0.72	8603	6161	10101
	$\mathbb{F}_{(2^{67})^2}$	3289	1206	1.34	6044	8085	17764
	$\mathbb{F}_{2^{139}}$	6718	973	0.80	9079	7303	11999
	$\mathbb{F}_{(2^{71})^2}$	3463	1278	1.49	6304	9367	20759
D=3	$\mathbb{F}_{(2^{67})^2}$	3468	1206	0.99	6224	6140	13279
	$\mathbb{F}_{(2^{71})^2}$	3647	1278	1.11	6489	7226	15764
D=4	$\mathbb{F}_{(2^{67})^2}$	3782	1206	0.83	6537	5406	11389
	$\mathbb{F}_{(2^{71})^2}$	3967	1278	0.91	6808	6199	13180

The designs were synthesized using Synopsis Design-analyzer for the frequency of 175 kHz and a 0.25 μm CMOS library. One of our main reasons for using composite fields was to reduce the ALU's area. This is clearly visible in Tables 3 and 4. We notice that the ALU varies in size from 2863 to 7379 gates and the smallest one is obtained for the field $\mathbb{F}_{(2^{67})^2}$, without the squarer and with a bit-serial multiplier. However, the performance is the worst for this case, requiring more than 2 seconds for one point multiplication. The total area without RAM includes the sizes of the ALU, the CU, the counter and the shift-register. The largest portion of that is occupied by the key register *i.e.* 1.4 and 1.5 kgates for fields $\mathbb{F}_{2^{131}}$ and $\mathbb{F}_{2^{139}}$, respectively. The control logic takes between 10% and 15% of a whole design.

In the last two columns, we computed the area-time product for two cases, including RAM and not including RAM. To map the number of bits to be stored to actual gates we used a factor of 6, which is conservative when using SRAM. If we were to use dedicated embedded RAM, it would be possible to half the area requirement (see for example [40, 23]), at the very least.

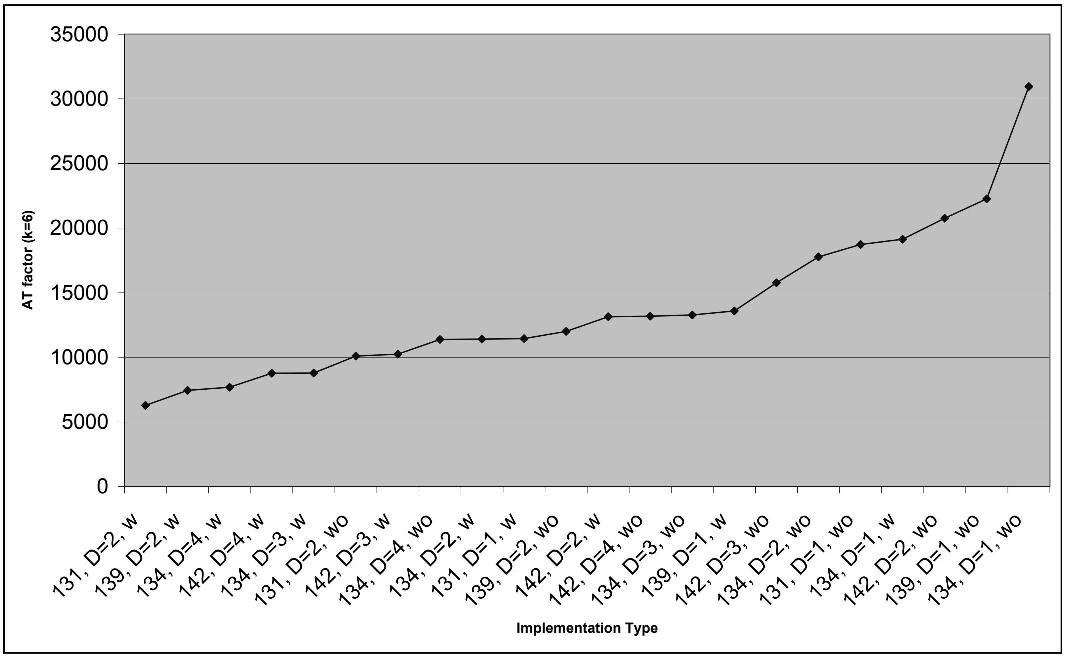

Fig. 4. The area time product assuming intermediate data saved in SRAM.

From Table 3 and looking at the AT-product values, we conclude that, in general, it is beneficial to use a digit-serial multiplier and a squarer. However, these options are not the most compact. For compactness one should choose an implementation without squarer. The total area is expressed without RAM for two reasons. First, it is hard to exactly map it to the corresponding number of gates and second, most tags have RAM available. Some high-end tags have therefore the possibility to store 1k bits, which would be enough in some cases presented in Table 3. Figure 4 gives the best cases for the AT factor where the area also includes the cost of RAM.

We compare our results with other related work in Table 5. It is hard to compare with other related work as there is no previous ECC implementation suitable for RFIDs. We chose here the architecture with the best timing although it is possible to have an adequate solution requiring a total of 13,646 gates with a performance that is still below 1 second (0.84 sec). We stress here again that we obtain these figures by including very conservative estimates for RAM in the total gate count. In fact, a RAM cell that requires 6 equivalent gates to be implemented is a register cell. A typical full-custom RAM cell requires somewhere between 1 and 2 equivalent gates, thus bringing the total area required for the design under 10,000 gates. Other optimizations involve the shift register for the key, which is of full length and requires 1.5 kgates. This can still be improved by loading the key in two or more parts, thus, reducing the area significantly.

8 Concluding Remarks

This work provides evidence that ECC on RFID might be a viable solution in the near future. This is important as it allows much more sophisticated protocols based on public-key cryptography than currently being considered for use in RFID. We investigated several options considering ECC over $\mathbb{F}_{2^p}$, p a prime, operands ranging between 130 and 140 bits in length, and composite fields. We also considered different ALU configurations to obtain more compact and still acceptable performance. We follow design criteria that would lead to low-power implementations, *i.e.* we try to minimize the area and reduce the operating frequency. The best architecture with respect to

Table 5. Performance and area of different algorithms and implementations

Source	Algorithm	Finite field/ Parameter Size	Area [$gates$]	Technology [μm]	Op. Frequency [kHz]	Performance [ms]
[15]	NTRUEncrypt	$N = 167, p = 3, q = 128$	3000	0.13	500	58.45
[31]	AES	block size = 128 bits	3595	0.35	100	10.2 (1016 cycles)
[26]	SHA-1	data size = 512 bits	4276	0.13	500	0.81 (405 cycles)
this work (smallest area)	EC	$\mathbb{F}_{(2^{67})^2}$	12,944	0.25	175	2.39 sec.
this work (smallest AT product, fastest)	EC	$\mathbb{F}_{2^{131}}$	14,735	0.25	175	430
[15]	EC	$\mathbb{F}_{p_{100}}$	18,720	0.13	500	410.5
[52]	EC	$\mathbb{F}_{2^{191}}$ and $\mathbb{F}_{p_{192}}$	23,000	0.35	68,500	6.7
[41]	EC	$\mathbb{F}_{p_{166}}$	30,333	0.13	20,000	31.9
[47]	EC	$\mathbb{F}_{(2^{89})^2}$	191,000	0.5	20,000	4.4

both area and performance is slightly larger than 10k gates. Future work will investigate the exact amount of power consumed by our processors, the cost of side-channel attack countermeasures, and concentrate on the further investigation of protocols based on public-key cryptography for RFID.

References

1. G. Avoine, E. Dysli, and P. Oechslin. Reducing Time Complexity in RFID Systems. In B. Preneel and S. E. Tavares, editors, *Selected Areas in Cryptography — SAC 2005*, volume LNCS 3897, pages 291–306. Springer, 2005.
2. S. Babbage, D. Catalano, L. Granboulan, A. Lenstra, C. Paar, J. Pelzl, T. Pornin, B. Preneel, M. Robshaw, A. Rupp, N. Smart, and M. Ward. ECRYPT Yearly Report on Algorithms and Keysizes (2004). Technical Report D.SPA.10, ECRYPT - European Network of Excellence in Crpytology, March 1st, 2005. Revision 1.0. Available at `http://www.ecrypt.eu.org/documents.html`.
3. T. Beth and D. Gollmann. Algorithm engineering for public key algorithm. *IEEE Journal on Selected Areas in Communications*, 7(4):458–465, May 1989.
4. M. Blaze, W. Diffie, R. L. Rivest, B. Schneier, T. Shimomura, E. Thompson, and M. Wiener. Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security - A Report by an Ad Hoc Group of Cryptographers and Computer Scientists. Available at `http://theory.lcs.mit.edu/~rivest/publications.html`, January 1996.
5. P. Bulens, G. M. de Dormale, and J.-J. Quisquater. "Hardware for Collision Search on Elliptic Curve over $GF(2^m)$". In *"Special-purpose Hardware for Attacking Cryptographic Systems — SHARCS'06"*, Cologne, Germany, April 03-04, 2006.
6. M. Ciet, J.-J. Quisquater, and F. Sica. A Secure Family of Composite Finite Fields Suitable for Fast Implementation of Elliptic Curve Cryptography. In C. Pandu Rangan and C. Ding, editors, *Progress in Cryptology — INDOCRYPT 2001*, volume LNCS 2247, pages 108–116. Springer, 2001.
7. Certicom Corp. Certicom ECC Challenge. Available at `http://www.certicom.com/index.php?action=res\,ecc_challenge`.
8. E. D. Mastrovito. *VLSI Architectures for Computation in Galois Fields.* PhD thesis, Dept. Electrical Engineering, Linköping University, Linköping, Sweeden, 1991.
9. M. Feldhofer, S. Dominikus, and J. Wolkerstorfer. Strong Authentication for RFID Systems Using the AES Algorithm. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems — CHES 2004*, volume LNCS 3156, pages 357–370. Springer, 2004.
10. International Organization for Standardization. ISO/IEC 18000-3. Information Technology AIDC Techniques - RFID for Item Management, March 2003.
11. 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.
12. S. D. Galbraith, F. Hess, and N. P. Smart. Extending the GHS Weil Descent Attack. In Lars R. Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, volume LNCS 2332, pages 29–44. Springer, 2002.

13. S. D. Galbraith and N. P. Smart. A Cryptographic Application of Weil Descent. In M. Walker, editor, *Cryptography and Coding — IMA Int. Conf.*, volume LNCS 1746, pages 191–200. Springer, 1999. The full version of the paper is HP Labs Technical Report,HPL-1999-70.
14. B. Gassend, D. E. Clarke, M. van Dijk, and S. Devadas. Silicon physical unknown functions. In V. Atluri, editor, *ACM Conference on Computer and Communications Security — CCS 2002*, pages 148–160. ACM, November 2002.
15. 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.
16. 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.
17. P. Gaudry. An Algorithm for Solving the Discrete Log Problem on Hyperelliptic Curves. In B. Preneel, editor, *Advances in Cryptology — EUROCRYPT 2000*, volume LNCS 1807, pages 19–34. Springer, 2000.
18. P. Gaudry, F. Hess, and N. P. Smart. Constructive and Destructive Facets of Weil Descent on Elliptic Curves. *Journal of Cryptology*, 15(1):19–46, 2002.
19. 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.
20. T. Güneysu, C. Paar, and J. Pelzl. "on the security of elliptic curve cryptosystems against attacks with special-purpose hardware". In *"Special-purpose Hardware for Attacking Cryptographic Systems — SHARCS'06"*, Cologne, Germany, April 03-04, 2006.
21. T. E. Güneysu. Efficient Hardware Architectures for Solving the Discrete Logarithm Problem on Elliptic Curves. Diplomarbeit, Chair for Communication Security — Ruhr-Universität Bochum, January 31st, 2006. Available at `http://www.crypto.rub.de/theses.html`.
22. F. Hess. The GHS Attack Revisited. In Eli Biham, editor, *Advances in Cryptology — EUROCRYPT 2003*, volume LNCS 2656, pages 374–387. Springer, 2003.
23. K. Itoh. Low-Voltage Embedded RAMs in the Nanometer Era. In *IEEE International Conference on Integrated Circuits and Technology — ICICT 2005*, pages 235–242. IEEE Computer Society, 2005.
24. M. Jacobson, A. Menezes, and A. Stein. Solving elliptic curve discrete logarithm problems using Weil descent. *Journal of the Ramanujan Mathematical Society*, 16:231–260, 2001.
25. A. Juels and S. A. Weis. Authenticating pervasive devices with human protocols. In V. Shoup, editor, *Advances in Cryptology: Proceedings of CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 293–308. Springer-Verlag, 2005.
26. J.-P. Kaps and B. Sunar. Energy comparison of AES and SHA-1 for ubiquitous computing. In E. Sha, editor, *IFIP International Conference on Embedded and Ubiquitous Computing — EUC 2006*, LNCS. Springer, 2006. To appear.
27. L. Song and K.K. Parhi. Low Energy Digit-Serial/Parallell Finite Field Multipliers. *Kluwer Journal of VLSI Signal Processing Systems*, 19(2):149–166, 1998.
28. A. K. Lenstra. Key Lengths. In Hossein Bidgoli, editor, *Handbook of Information Security*. Wiley Publishing, To appear. Electronically published on June 30th, 2004. Available at `http://cm.bell-labs.com/who/akl/index.html`.
29. A. K. Lenstra and E. Verheul. Selecting Cryptographic Key Sizes. *Journal of Cryptology*, 14(4):255–293, December 2001.
30. 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.
31. 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.
32. M. Maurer, A. Menezes, and E. Teske. Analysis of the GHS Weil descent attack on the ECDLP over characteristic two finite fields of composite degree. *LMS Journal of Computation and Mathematics*, 5:127–174, 2002.
33. A. Menezes and M. Qu. Analysis of the Weil Descent Attack of Gaudry, Hess and Smart. In D. Naccache, editor, *Topics in Cryptology — CT-RSA 2001*, volume LNCS 2020, pages 308–318, 2001.
34. A. Menezes and E. Teske. Cryptographic implications of Hess' generalized GHS attack. *Applicable Algebra in Engineering, Communication and Computing*, 16(6):439–460, 2006.
35. A. Menezes, E. Teske, and A. Weng. Weak Fields for ECC. In T. Okamoto, editor, *Topics in Cryptology — CT-RSA 2004*, volume LNCS 2964, pages 366–386. Springer, 2004.
36. A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1997.

37. V. S. Miller. Use of elliptic curves in cryptography. In H. C. Williams, editor, *Advances in Cryptology – CRYPTO '85*, volume 0218 of *LNCS*, pages 417–426. Springer-Verlag, 1986.
38. P. Montgomery. Speeding the Pollard and Elliptic Curve Methods of Factorization. *Mathematics of Computation*, Vol. 48:243–264, 1987.
39. D. Naccache, N. P. Smart, and J. Stern. Projective coordinates leak. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology — EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 257–267. Springer, 2004.
40. Y. Nakagome, M. Horiguchi, T. Kawahara, and K. Itoh. Review and future prospects of low-voltage RAM circuits. *IBM Journal of Research and Development*, 47(5/6):525–552, 2003.
41. E. Özturk, B. Sunar, and E. Savaş. Low-Power Elliptic Curve Cryptography Using Scaled Modular Arithmetic. 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 92–106. Springer-Verlag, 2004.
42. B. Skoric P. Tuyls. Secret key generation from classical physics. *Philips Research Book Series*, September 2005.
43. R. Pappu. Physical one-way functions. *Science*, 297(6):2026, 2002.
44. J. M. Pollard. Monte Carlo methods for index computation (mod p). *Mathematics of Computation*, 32:918–924, 1978.
45. PriceWatch.info. Price History of Athlon 3200+. Available at `http://www.pricewatch.info/item/16909`.
46. 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.
47. R. Schroeppel, C. L. Beaver, R. Gonzales, R. Miller, and T. Draelos. A low-power design for an elliptic curve digital signature chip. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2002*, volume LNCS 2523, pages 366–380, 2002.
48. N. P. Smart. How Secure Are Elliptic Curves over Composite Extension Fields? In B. Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, volume LNCS 2045, pages 30–39. Springer, 2001.
49. P. Tuyls and L. Batina. RFID-tags for Anti-Counterfeiting. In D. Pointcheval, editor, *Topics in Cryptology - CT-RSA 2006*, Lecture Notes in Computer Science, San Jose, USA, February 13-17 2006. Springer Verlag.
50. P. C. van Oorschot and M. J. Wiener. Parallel Collision Search with Cryptanalytic Applications. *Journal of Cryptology*, 12(1):1–28, 1999.
51. J. Wolkerstorfer. Is Elliptic-Curve Cryptography Suitable to Secure RFID Tags?, 2005. Workshop on RFID and Lightweight Crypto, Graz, Austria.
52. J. Wolkerstorfer. Scaling ECC Hardware to a Minimum. In ECRYPT workshop - Cryptographic Advances in Secure Hardware - CRASH 2005, September 6-7 2005. invited talk.
53. Huapeng Wu. Bit-parallel finite field multiplier and squarer using polynomial basis. *IEEE Trans. Computers*, 51(7):750–758, 2002.

A History of Attacks on Composite Fields

Table 6. Composite field degrees considered insecure and history of related attacks.

Year	Source	History of attacks
1998	[11]	Weil Descent Attack introduced. Fields $\mathbb{F}_{2^{k \cdot m}}$ are potentially weak
1999	[13]	Attack applied to EC over $\mathbb{F}_{2^{4 \cdot n}}$
2000	[17]	Algorithm for attacking hyperelliptic curves of genus > 4 with complexity better than Pollard's Rho
2001	[33]	Weil Descent attack against EC over fields $\mathbb{F}_{2^p}$, p prime is infeasible. Only a small fraction of EC over $\mathbb{F}_{2^{155}}$ are susceptible to the Weil Descent attack
	[48]	It is shown that curves over $\mathbb{F}_{2^{4 \cdot n}}$ should not be used and that curves over $\mathbb{F}_{2^{5 \cdot n}}$ are still secure but attack using Weil Descent method offers better complexity than Rho method
	[24]	Weil Descent attack is shown to work on curves defined over $\mathbb{F}_{2^{62}}$, $\mathbb{F}_{2^{93}}$, $\mathbb{F}_{2^{124}}$, $\mathbb{F}_{2^{155}}$. Attack on curves over $\mathbb{F}_{2^{155}}$ is only applicable to insignificant fraction of curves.
	[6]	Based on the analysis method of [33], it is shown that fields $\mathbb{F}_{2^{2 \cdot p}}$, p prime are not susceptible to the Weil Descent attack. Specific instances are $\mathbb{F}_{2^{178}}$, $\mathbb{F}_{2^{226}}$, $\mathbb{F}_{2^{1018}}$ and $\mathbb{F}_{2^{1186}}$.
2002	[18]	Very efficient algorithm to reduce the ECDLP to the DL problem in a Jacobian of a hyperelliptic curve over $\mathbb{F}_q$. Index calculus method to solve the DL problem on hyperelliptic curves of genus ≥ 4.
	[32]	Shown that the Weil Descent attack is not applicable to ANSI X9.62 Standard curves $\mathbb{F}_{2^{176}}$, $\mathbb{F}_{2^{208}}$ and $\mathbb{F}_{2^{272}}$, $\mathbb{F}_{2^{304}}$, and $\mathbb{F}_{2^{368}}$. However, if efficient algorithm is found to compute isogenous curves from among most vulnerable ones, the Weil Descent attack yields better complexity than the Rho method
	[12]	The attack from [18] to a much larger number of elliptic curves over certain composite fields of even characteristic. Larger proportion than previously thought of EC over $\mathbb{F}_{2^{155}}$ should be considered weak.
2003	[22]	Further generalization of [18]. Larger number of EC curves defined over $\mathbb{F}_{2^{155}}$ to attack.
2004	[35]	It is shown that EC defined over fields $\mathbb{F}_{2^{5 \cdot k}}$ are weak, in the sense that Weil Descent attacks are faster than Pollard's Rho ones. In particular, curves over $\mathbb{F}_{2^{210}}$ can be solved a factor of 2^{13} faster than with Pollard's Rho (and for one quarter of these curves 2^{20} times faster). EC over fields $\mathbb{F}_{2^{4 \cdot k}}$ are weak but not as weak as those defined over $\mathbb{F}_{2^{5 \cdot k}}$
2006	[34]	Analysis strongly suggests that finite fields $\mathbb{F}_{2^n}$ where n is divisible by 3, 5, 6, 7 or 8, should not be used to implement EC cryptographic protocols.