Available online at www.sciencedirect.com
ScienceDirect
The Journal of Systems and Software 81 (2008) 1240–1247
The Journal of Systems and Software
www.elsevier.com/locate/jss

A pairing SW implementation for Smart-Cards

Guido M. Bertoni [b], Luca Breveglieri [a], Liqun Chen [c], Pasqualina Fragneto [b], Keith A. Harrison [c], Gerardo Pelosi [a,*]

[a] *Politecnico di Milano, P.zza L. Da Vinci, I-20133 Milano, Italy*
[b] *STMicroelectronics, Centro Direzionale Colleoni, I-20041 Agrate, Italy*
[c] *HPLabs, Filton Road BS34 8QZ Bristol, United Kingdom*

Received 29 October 2006; received in revised form 12 July 2007; accepted 25 September 2007
Available online 1 October 2007

Abstract

The aim of this work is to show the feasibility of the primitives of the identity based cryptosystems for applications in Smart-Cards. Several observations are applied to easily choose many supersingular elliptic curves over a prime field $\mathbb{F}_p$, $p > 3$, $p \equiv 3 \ mod \ 4$, in such a way that the size of the torsion subgroup, the curve order and the finite field characteristic are of minimal Hamming weight. We modify the Chudnovsky elliptic curve point representation to settle a dedicated coordinate system for pairings and to minimize the number of operations in the finite field. The encouraging timing results obtained for ST22 Smart-Card architecture show the feasibility of pairing primitives for embedded devices.

Keywords: Tate pairing; Elliptic curves; Software implementation

1. Introduction

In the early 90s, pairings were used to show that some classes of elliptic curves were unsafe for cryptographic purposes because by using the Weil pairing (as shown by Menezes et al. (1993)) or the Tate pairing (as shown by Frey et al. (1999)), one can reduce the discrete logarithm problem on these elliptic curves to the discrete logarithm problem in a finite field. In 2000, Joux's tripartite Diffie–Hellman key agreement protocol gave a positive usage of the pairings (Joux, 2000). This use of the pairings has inspired much research to find novel cryptographic protocols and to improve the existing ones. There is now a wide range of public key primitives that rely on the use of pairing functions. The most significant outcome of using pairings is pairing-based identity-based cryptography. The concept of identity-based cryptography is due to Shamir (1984), where a public key is derived from publicly identifiable information such as an e-mail address, and the corresponding private key is created by binding the identity with a trusted authority's master secret key. This idea avoids the reliance on certificates to validate the authenticity of a public key and, in some situations, simplifies the infrastructure of a public key system. Shamir proposed an identity-based signature scheme, but left building identity-based encryption as an open problem. A number of solutions for implementing this idea were presented almost 20 years later by Boneh and Franklin (2001), Cocks (2001) and Sakai et al. (2000). Apart from the Cocks scheme, both the Boneh–Franklin scheme and the Sakai et al. scheme are using pairings. In addition, there are many other cryptographic protocols benefiting from using pairings, including a variety of signature schemes, e.g. short signatures (Boneh et al., 2001), blind signatures, ring signatures (Zhang and Kim, 2002), group signatures (Boneh et al., 2004) and so on; a variety of key-establishment schemes, e.g., identity-based key agreement, authenticated tripartite key agreement, etc.; many other applications, such as signcryption, identity-based signcryption, authentication and identification.

* Corresponding author.
E-mail address: pelosi@elet.polimi.it (G. Pelosi).

0164-1212/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2007.09.022

A survey of pairing-based cryptographic protocols can be found in Dutta et al. (2004), while for further deepening and latest developments the reader is pointed to International Association for Cryptologic Research (2007), Barreto (2007).

One of the most demanding benefits of modern Smart-Cards is the ability to support cryptographic protocols. Smart-Cards are portable computing devices which are leaving their past role as mere carriers of confidential information to become more and more sophisticated embedded platforms. By implementing identity-based features on a Smart-Card, it is possible to envision practical and cost-effective online and off-line secure business/commercial communication solutions in various areas which could embrace the wireless management of secure documents, personal-authorization tokens, electronics and mobile commerce. The pairing-based paradigm, as mentioned above, enables to turn a simple, fully recognized identity or role into a public/private key pair. Such possibility allows for one of the communication parties, e.g. the receiver, to dynamically change the link between the identity and the role of the user without impacting the other party, e.g. the sender (Mont et al., 2003).

The core of any practical identity based scheme is based on the implementation of a pairing function. At the current state of the art, the Tate pairing is considered the most convenient pairing function in terms of computational cost. As the security standards for public key cryptosystems will increase, the task to choose the most performing pairing algorithm with security equivalent to 128-, 192-, or 256-bit AES keys, become more tricky. Koblitz and Menezes (2005) point out how the implementation advantages in adopting an elliptic curve with embedding degree $k = 2$ defined over a prime field, will fall back on the Tate Pairing algorithm until its equivalent security turn to be from 80-bit to 192-bit; afterwards it will get on to the implementation of Weil pairing for higher security requirements.

In the following of the paper we discuss and evaluate a software implementation of the Tate pairing based on supersingular elliptic curves defined over $\mathbb{F}_p$ to be used in a specialized software library module for the proprietary 32-bit Smart-Card platform ST22 by STMicroelectronics. The encouraging timing results allow to assert the feasibility of pairing-based primitives also on embedded devices, despite their high computational complexity. The most recent timing results reported from the literature favorably compares with our implementation, as well.

The rest of the paper is organized as follows: Section 2 gives the picture of the necessary mathematical preliminaries that one needs to fully understand the subsequent sections. Section 3 tackles the topic of efficient implementation and describes the methodology followed to implement the BKLS algorithm, while Section 4 reports the timing results obtained w.r.t. the target platform and compare with other related works on pairing SW implementation for constrained architectures. Finally, Section 5 points out the concluding remarks about the discussed work.

2. Preliminaries on pairings

The milestone for all the practical deployments of identity-based protocols is the existence of bilinear maps called pairings, which are mathematically defined in terms of the elliptic curves algebra with coefficients in a finite field. However, from a functional point of view the necessity of such a function could be best addressed considering the problem of assigning a secret between two interlocutors to establish a confidential communication over an insecure channel. The Diffie–Hellman protocol represents the best known solution to such a problem and uses the arithmetic in the multiplicative subgroups of finite fields. An alternative scenario involves a centralized entity that defines a secret and associates to each user a token built as the output of a known one-way function. Such a function combines the centralized secret and the user's public identification. The possibility to establish a common secret exclusively shared by any pair of users is only guaranteed if there is a black box able to securely blend the tokens of the two users to produce another one-way output value. This additional value should be a function of the centralized secret (which remains always unknown and un-computable to all users) and of the two users' identifications.

To be just a little more accurate, a pairing can be seen as a bi-variate function:

$$e : G_1 \times G_1 \rightarrow G_2$$

where G_1 and G_2 are finite cyclic groups with an additive and multiplicative law of composition, respectively, and of the same order.

The main property that should be satisfied by the above map is a peculiar multiplicative law, called *bilinearity*, with the characteristics to be: efficiently computable, distributive to the sum of G_1 and with a computationally hard discrete logarithm problem both in G_1 and in G_2. Let $P,Q,P_1,P_2,Q_1,Q_2 \in G_1$ then $e(P_1 + P_2, q) = e(P_1, Q)e(P_2, Q)$, $e(P, Q_1 + Q_2) = e(P, Q_1)e(P, Q_2)$ in such a way, given an integer r, the iterated sum rP in G_1 is translated as $e(rP,Q) = e(P,rQ) = e(P,Q)^r$.

Practical cryptosystems use pairings to map the discrete logarithm problem (DLP) from the abelian groups of an elliptic curve G_1 to the multiplicative subgroup G_2 of some extension of the finite field on which the elliptic curve is defined.

Let E be an elliptic curve over a finite field F_q and r an integer co-prime with the field characteristic and such that $r|\#E(\mathbb{F}_q)$, and consider the minimum field extension $\mathbb{F}_{q^k}$ such that $r|q^k - 1$. Therefore, such field include the multiplicative subgroup of r-th roots of unity. Therefore assume $G_2 = \mu_r = \{u \in \mathbb{F}_{q^k} : u^r = 1\}$.

Assume G_1 to be the group of the r-torsion points of the curve, i.e. $G_1 = E(\mathbb{F}_q)[r] = \{P \in E : [r]P = \mathcal{O}\}$. As long as

$r \nmid q-1$, the group $E[r]$ is completely included in $E(\mathbb{F}_{q^k})$ if and only if $r|q^k-1$ (Balasubramanian and Koblitz, 1998).

In order to efficiently evaluate the pairing, the embedding degree k should be sufficiently large without compromising effective implementation. There is a trade-off between the efficient computation of the pairing and the guarantee of the hardness of the DLP in $\mathbb{F}_{q^k}$ (k large). In Balasubramanian and Koblitz (1998) it is shown that a randomly selected elliptic curve is not suitable for pairings-based cryptography because of the large value of the *embedding degree* k ($k > \log(q)$). On the other side, there is a number of algorithms to find out pairing-friendly curves (k not too large) (Barreto et al., 2002b; Galbraith et al., 2004; Scott and Barreto, 2006).

The *Tate pairing* is well-defined, non-degenerate, bilinear pairing assuming the above stated groups G_1, G_2. A restricted version of the pairing which retains all of these properties and is usefully employed for cryptographic applications is subsequently described.

The mathematical Tate pairing is defined in terms of rational functions over points of an elliptic curves evaluated in a divisor (Blake et al., 2006; Silverman, 1994).

The definition of *divisor* can be stated as a finite formal sum of elliptic curve points, i.e.

$$D = \sum_{P \in E} m_i(P_i) \quad \text{where} \quad m_i > 0$$

The divisor associated to a rational function defined over the points of an elliptic curve is defined as the formal sum of its zeros (positive sign) and poles (negative signs) counted with their multiplicity. More definition and mathematical details about rational functions defined on elliptic curves and divisor theory can be found in Blake et al. (2006), Silverman (1994).

A divisor D is named *principal* if there exists a rational function $f \in \mathbb{F}_q(E)$ such that $D = div(f)$; besides a divisor $D = \sum_{P \in E} m_i(P_i)$ is principal if and only if $\sum_i m_i = 0$ and $\sum_{P_i \in D}[m_i]P = \mathcal{O}$ (Silverman, 1994) (The point at infinity $\mathcal{O}$ is the neutral element of curve additive group of points).

To evaluate a rational function $f \in \mathbb{F}_q(E)$ in a divisor $D = \sum_{P \in E} m_i(P_i)$ it is necessary to introduce the following definition: $f(D) = \prod_{P_i \in E} f(P)^{m_i}$. The only restriction is that D and the divisor of f do not share any common points.

Let $P \in E(\mathbb{F}_q)[r]$, then $r(P) - r(\mathcal{O})$ is a principal divisor. So there is a rational function $f_P \in \mathbb{F}_{q^k}(E)$ with $div(f_P) = r(P) - r(\mathcal{O})$. Let $Q \in E(\mathbb{F}_{q^k})[r]$ be a point with coordinates in $\mathbb{F}_{q^k}$ and consider the divisor $D_Q \in Div^0(E)$ such that $D_Q \sim (Q) - (\mathcal{O})$ with disjoint support from that of f_P.

The reduced Tate pairing definition can be stated as follows:

$$e : E(\mathbb{F}_q)[r] \times E(\mathbb{F}_{q^k})[r] \to \mu_r;$$
$$e(P,Q) = f_P(D_Q)^{(q^k-1)/r}$$

An effective algorithm for computing a rational function f_P used in the previous definition was conceived by Miller (1986). The algorithm implements a double-and-add strategy with some extra computation due to the construction and evaluation of the rational function f_P by means of the equations of straight lines (e.g. $g_{U,V}(Q) = \frac{y_V - y_U}{x_V - x_U}(x_Q - x_U) + y_U - y_Q$, $U, V, Q \in E(\mathbb{F}_q)$).

Let $f_{i,P}$ be a function such that $div(f_{i,P}) = ([i]P) - i(P) + (i-1)(\mathcal{O})$. Starting from $f_{1,P} = 1$, the algorithm uses an addition chain for $[r]P$ to compute the value $f_{r,P}(Q) = f_P(Q)$ (Miller, 1986).

Much research effort has been aimed to the optimization of the algorithm for pairing computation on *supersingular* curves (Barreto et al., 2002a; Galbraith et al., 2002), specializing algorithmic variants depending on the algebraic properties of finite fields with different characteristic (Duursma and Lee, 2003; Barreto et al., 2004).

An elliptic curve is said to be *supersingular* if it does not have any points with order that is a multiple of the finite field characteristic. Moreover such curves have the property to exhibit an embedding degree k which is always less than 6.

In the current paper we restrict ourselves to discussion of these curves defined over finite fields with prime characteristic $E(\mathbb{F}_p)$.

One of the most important properties of such curves is the existence of a distortion map $\phi : E(\mathbb{F}_q) \to E(\mathbb{F}_{q^k})$ (Verheul, 2001). A distortion map is used to map an r-torsion point from a base field to an r-torsion point in an extension field. In such a way the definition of the Tate pairing can be modified as follows, including the optimizations highlighted in Barreto et al. (2002a):

$$e(P,Q) = f_P(\phi(Q))^{(q^k-1)/r}$$

The previous formula always lead to a well-defined, non-degenerate pairing definition $e(P,\phi(Q))$, where $e(P,\phi(Q)) \neq 1 \forall P, Q, \in E(\mathbb{F}_p)[r]$. The coordinates of both P and Q are elements of the base field; therefore P and $\phi(Q)$ results to be linearly independent.

3. Computing the tate pairing

In this work, we will describe an implementation of Tate pairing for supersingular curves defined over prime order finite field $\mathbb{F}_p$. Because of the prime characteristic p the following description will be based on the so-called BKLS algorithm (Barreto et al., 2002a).

Most recent developments centered on the derivation of an alternative pairing definition for non-supersingular curves an known as *Ate pairing* (Hess et al., 2006), is not considered.

One of the prevalent reasons because of our choice about supersingular curve is for the convenient way to choose the parameters for computing the BKLS algorithm. We envisioned an easy procedure to compute a large number of values for the characteristic p, the group order r and the final exponent with the aim to trade off the best combination derivable from the applied optimizations.

When working in characteristic p, there are two possible choices for the equation of the corresponding elliptic curves:

$$E_1 : y^2 = x^3 + x \quad \text{over} \quad \mathbb{F}_p, \quad p \equiv 3 \bmod 4,$$
$$\phi(x,y) = (-x, iy) \quad \text{where} \quad i^2 \equiv -1$$
$$E_2 : y^2 = x^3 + a \quad \text{over} \quad \mathbb{F}_p, \quad p \equiv 2 \bmod 3,$$
$$\phi(x,y) = (\xi x, y) \quad \text{where} \quad \xi^3 = 1$$

Most of the optimizations given below only work for curve E_1 – consequently we will not consider curve E_2 any further. For such finite fields the pairing function for points $P,Q \in E(F_p)[r]$ assumes the value $e_r(P,\phi(Q)) \in \mathbb{F}_{p^2}$. Having $p \equiv 3 \bmod 4$, the arithmetic of the quadratic extension field $\mathbb{F}_{p^2}$ is most efficiently managed in polynomial base: $\mathbb{F}_{p^2} \equiv \mathbb{F}_p(\alpha) = \{a_1\alpha + a_0, a_i \in \mathbb{F}_p, \alpha = \sqrt{-1}\} \equiv \mathbb{F}_p[x]/(x^2+1)$. The adaption of BKLS algorithm to the current case study results in Algorithm 1.

Algorithm 1 BKLS Algorithm (Barreto et al., 2002a)

Require $t = \lceil \log_2(r) \rceil$, $r = (r_{t-1}, \ldots, r_0)_2, P, Q \in E(\mathbb{F}_p)[r]$

Ensure $e(P,Q) = f_P(\phi(Q))^{\frac{p^2-1}{r}} \in \mathbb{F}^*_{p^2}$

1: $f \leftarrow 1$
2: $V \leftarrow P$
3: **for** $i \leftarrow t-2$ **down to** 1 **do**
4: $\quad f \leftarrow f^2 \cdot g_{V,V}(\phi(Q))$
5: $\quad V \leftarrow 2V$
6: $\quad$ **if** $r_i == 1$ **then**
7: $\quad\quad f \leftarrow f \cdot g_{V,P}(\phi(Q))$
8: $\quad\quad V \leftarrow V + P$
9: $\quad$ **end if**
10: $\quad f \leftarrow f^2 \cdot g_{V,V}(\phi(Q))$
11: **end for**
12: $f \leftarrow f^{\frac{p^2-1}{r}}$

The use of a distortion map makes it possible to avoid the evaluations of the denominators provided in the Miller Algorithm, because their values exponentiate to the unity after the final exponentiation. Besides, making use of the same argument, in the last iteration ($i = 0$) of the algorithm the evaluation of $g_{V,P}(\phi(Q))$ is avoided because $V + P = \mathcal{O}$.

3.1. Parameters generation

The choice to select a supersingular elliptic curve allows to work out an efficient procedure to select a large set of parameters to further speed up the implementation of Algorithm 1. The selected elliptic curve have an almost prime order: $\#E = p + 1 = rc$, where the prime r is the order of the torsion group over the elliptic curve, while the co-factor integer c appears in the final exponentiation since the following equality holds: $\frac{p^2-1}{r} = (p-1)c$. The expression of $\#E$ explicitly reveals the influence of such value on the finite field arithmetic optimizations, and on the operations involved in the computation of the final exponentiation.

To guarantee a security level comparable to 1024-bit RSA it should be assured that $r \geqslant 2^{160}$ and $p \geqslant 2^{512}$ hold, assuming that the DLP over F_{p^2} is as hard as that over an F_p field of double bit size. Having $p \equiv 3 \bmod 4$ ($\#E \equiv 0 \bmod 4$), it is required that the cofactor be a multiple of 4.

The efficiency of Miller Algorithm depends on the hamming weight of the prime r. Thus, it is desirable that r is a prime with low hamming weight-ideally, 3. Besides, also the final exponentiation would benefit of a low hamming weight for the cofactor c. On the other hand having the characteristic p with a low hamming weight permits the modular arithmetic to be optimized.

The previous considerations lead to perform a search procedure that fixes a low hamming weight prime r of about 160 bits and then repeatedly tries low Hamming weight values c of about 352 bits, with $c \equiv 4 \bmod 8$, until a prime $p = rc - 1$ is found; consequently it will have a low Hamming weight. Such procedure allows us to quickly find a large number of parameters with a sufficiently small Hamming weight (in less than a few seconds). The Hamming weight of a typical tuple computed as above is $hw(l,c,u,p) = (3,3,9,10)$.

3.2. Eliminating the hidden inversions

Looking in more detail at the computations within BKLS algorithm – namely, point addition, point doubling and the computation of the straight lines function $g_{V,P}$ and $g_{V,V}$ – there are several recognizable field inversions. In software implementations the cost ratio of modular inversion versus modular multiplication typically ranges from 9 to 30 (Cohen et al., 1998); hence, it is necessary to compute as few as possible inversions and, at the same time, jointly optimize the entire block of operations executed in the body loop of the algorithm.

The points of an elliptic curve can be represented by a number of different coordinate systems. For each of them, the cost of the group operations depends on the type and the number of involved field operations, as shown in Table 1.

The Jacobian coordinates offer a faster doubling but a slower addition than the homogeneous ones, while the Chudnovsky formulas (Chudnovsky and Chudnovsky, 1986) differ from the Jacobian ones only for the internal representation of the point, and reduce the computation time of an addition but slightly increase the doubling time.

As in the BKLS algorithm the selected order r has a low Hamming weight, in order to minimize the number of

Table 1
Cost of group operations over $E(\mathbb{F}_p)$ in different coordinate systems. (I = inversion, M = multiplication, S = squaring)

Coordinates	Representation	Doubling	Addition
Affine	$(X; Y)$	I + 2M + 2S	I + 2M + S
Homogeneous	$(X; Y; Z)$	7M + 5S	12M + 2S
Jacobian	$(X; Y; Z; Z^2)$	4M + 6S	12M + 4S
Chudnovsky	$(X; Y; Z; Z^2; Z^3)$	5M + 6S	11M + 3S

operations in the base field we set up a specific coordinate system for the pairing computation, based on a modified version of the Chudnovsky coordinates $(X; Y; Z; W; V)$, where $W = Z^2$ and $V = X - WX_{\phi(Q)} = X + WX_Q$.

The adoption of such a hybrid coordinate system leads to express the group law formulas similar to the Jacobian ones, assuring the fastest doubling. We used Affine or Jacobian coordinates for the points P,Q or the temporary point V, respectively, maximizing the number of shared intermediate operations.

Given $P = (x_P, y_P, 1, 1, x_P + x_Q)$, $V = (x_1, y_1, z_1, w_1, v_1)$, $2V = (x_2, y_2, z_2, w_2, v_2)$ and $V + P = (x_3, y_3, z_3, w_3, v_3)$, the explicit formulas are shown in Tables 2 and 3. The computational cost falls to 10 multiplications plus 9 squarings in F_p for the doubling step of the BKLS algorithm, and to 15 multiplications plus 3 squarings for the addition step. The BKLS algorithm executes only $hw(r) - 2$ addition steps and $\lceil log_2 r \rceil$ doubling steps, thus the cost of the entire computation is substantially equal to that of the executed doubling steps, plus the cost of the final exponentiation.

3.3. Implementation of finite field arithmetic

The execution of BKLS algorithm requires both calculations in $\mathbb{F}_p$, and in $\mathbb{F}_{p^2}$ for the evaluation of formulas shown in the previous section and for the complete computation of the pairing value (Algorithm 1 – lines 4,10,12), respectively.

Table 2
Doubling formulas with modified Chudnovsky coordinates

$t_1 = 3x_1^2 + w_1^2$
$t_2 = 2y_1^2$
$t_3 = 2x_1 t_2$
$t_4 = 2t_2^2$

$x_2 = t_1^2 - 2t_3$
$y_2 = t_1(t_3 - x_2) - t_4$
$z_2 = 2y_1 z_1$
$w_2 = z_2^2$
$v_2 = x_2 + w_2 x_Q$

$g_{V,V}(\phi(Q)) = -v_1 t_1 + t_2 + \mathrm{i}(-z_2 w_1 y_Q)$

Table 3
Addition formulas with modified Chudnovsky coordinates

$t_1 = w_1 x_P$
$t_2 = x_1 - t_1$
$t_3 = t_2^2$
$t_4 = w_1 z_1 y_P$

$t_5 = y_1 - t_4$
$t_6 = (x_1 + t_1) t_3$
$t_7 = y_1 + t_4$
$t_8 = t_6 - 2x_3$

$x_3 = t_5^2 - t_6$
$2y_3 = t_5 t_8 - t_2 t_3 t_7$
$z_3 = z_1 t_2$
$w_3 = z_3^2$
$v_3 = x_3 + w_3 x_Q$

$g_{V,P}(\phi(Q)) = z_3 y_P - t_5(x_Q + x_P) + \mathrm{i}(-z_3 y_Q)$

Multiplications and squarings in $\mathbb{F}_{p^2}$ are computed using the well known Karatsuba tricks, both at the cost of three multiplications and squarings in the base field $\mathbb{F}_p$, respectively. Moreover, also the final exponentiation can be expressed in terms of operations in the base field. Having $f^{\frac{p^2-1}{r}} = f^{(p-1)c}$, with $f = a + ib \in \mathbb{F}_{p^2}$, the following easy derivation is possible

$$f^{(p-1)c} = \left(\frac{a - \mathrm{i}b}{a + \mathrm{i}b}\right)^c = \left(\frac{a^2 - b^2 + \mathrm{i}\left((a-b)^2 - (a^2 + b^2)\right)}{a^2 + b^2}\right)$$

revealing a total cost of 3 squares, one inversion and two multiplications in the base field. The only expensive item remains the final exponentiation by c to be executed in the arithmetic of $\mathbb{F}_{p^2}$.

On the basis of the description depicted so far, the implementation of BKLS algorithm requires only one other modular inversion for the computation of the final exponentiation. Hence, efficiency is not an issue and a traditional extended GCD algorithm (Menezes et al., 1996) can be used.

Modular multiplication was implemented using a *multiply-&-reduce* strategy with the reduction algorithm customized according to the selected modulus p.

The multiplication between two multi-precision numbers has been developed using a hybrid strategy that combines both the row-wise and column-wise multiplication techniques (Gura et al., 2004), taking advantage of a vector multiplication primitive that multiplies two words of the multiplicand by the entire multiplier. The partial products are reordered and accumulated to give the final result using the minimum number of registers and memory accesses.

As far as modular reduction is concerned, sophisticated techniques, like the Karatsuba algorithm (Karatsuba and Ofman, 1962) and the Montgomery reduction (Montgomery, 1985) have proven do not add any benefit, because of the not so large size of the operands currently used for public-key cryptosystems (512–2048-bit). The access to a fast native instruction for 32×32-bit multiplication with a full 64-bit result has revealed fundamental to adopt the multiply-&-reduce strategy as the best solution.

Since, all arithmetic is being performed mod p, the used modular reduction algorithm must be efficient. Traditionally, implementers of elliptic curve cryptography have used Solinas primes (Solinas, 1999) for the modular arithmetic of a prime order finite field. These lead to particularly efficient reduction algorithms. For the derivation discussed in the current paper, such an option is not available as the prime p had to satisfy other criteria. However, advantage of the fact that p has a low ($\sim$10) hamming weight can be taken.

The special form of the modulus, allows to customize the classical division algorithm to perform a fast quotient estimation. In such a way the reduction can be implemented without multiplications or divisions.

For example, assume to use the following parameters, with a processor word length of $w = 32$ bits:

$$r = 1 + 2^{17} + 2^{159}$$
$$c = 2^2 + 2^{191} + 2^{352}$$
$$p = 1 + 2^1 + 2^{19} + 2^{161} + 2^{191} + 2^{208} + 2^{350} + 2^{352} + 2^{369} + 2^{511}$$

when it comes to multiply two reduced values $z = xy \in \mathbb{F}_p$ ($p^2 < 2^{1023}$), working a word at a time, as in the traditional shift and subtraction algorithm, allows to use only $(\lceil \log_2(p) \rceil / w)^2$ single precision multiplications to perform the entire modular multi-precision multiplication. The implemented reduction strategy is described by Algorithm 2.

Algorithm 2 Customized Reduction Algorithm

Require $w = 32$, $b = 2^w$, $m = \lceil \log_2(p) \rceil / w$, $z = (z_{2m-1}, \ldots, m_0)_2$, $p = (p_{m-1}, \ldots, p_0)_2$ with $p_{m-1} = 0x8000000$
Ensure the residue $\bar{z} = (z_{m-1}, \ldots, z_0)_2$
1: $i \leftarrow 2m - 1$ **down to** m **do**
2: **if** $z_i == 0x8000000$ **then**
3: $q \leftarrow b - 1$
4: **else**
5: $q \leftarrow \lfloor (z_i b + z_{i-1})/2^{31} \rfloor$
6: **end if**
7: $z \leftarrow z - qpb^{i-m}$
8: **if** $z < 0$ **then**
9: $z \leftarrow z + pb^{i-m}$
10: **end if**
11: **end for**

The only expensive operation in Algorithm 2 is $z \leftarrow z - qpb^{i-m}$, where the modulus is expressed as $p = 2^{a_1} + 2^{a_2} + \ldots + 2^{a_n}$, $b = 2^{32}$, and q is a one word long variable. Having fixed the value of p, a procedure can be hard coded to perform this reduction step without the need for multiplication:

$$z \leftarrow z - (q \ll a_1 + 32(i-m)) - (a_2 + 32(i-m)) - \ldots - (a_n + 32(i-m))$$

An algorithm for the automatic generation of the entire multiplication code is also quite simply to envision.

4. Smart-Card implementation

The SmartJ ST22 platform (STMicroelectronics, 2005a) is a commercial multi-application Smart-Card that combines execution of Java bytecodes directly translated into native microcode instructions via a hardware decoder, with a proprietary native RISC instruction set. The native RISC mode copes with cryptographic and operating system support operations.

The processor core is a full 32-bit four-stage pipeline RISC architecture, with 32-bit data paths and sixteen 32-bit registers as well as a number of special-purpose registers with various lengths. The core is complemented by on-chip ROM, SRAM and up to 128K Bytes of EEPROM, as well as a set of standard peripheral circuits and custom plug-in circuits. A hardware memory protection unit provides highly secure control over how a program can access various regions of memory, while other built-in mechanisms protect against external physical attacks.

Separate memory buses, one to the on-chip SRAM and one to the ROM and EEPROM arrays, are provided. This allows data and instruction accesses to be overlapped, which provides faster execution since the great majority of data accesses are typically to the stack in SRAM while instructions are stored in ROM or EEPROM.

For implementing cryptographic algorithms, including Public Key and Secret Key types, an embedded library of specialized mathematical functions is provided. The discussion developed throughout the current work has lead to the upgrade of this software library.[1]

Table 4
Execution time of cryptographic primitives on the ST22 Smart-Card @ 33 MHz

Primitive	Time [ms]
Pairing	500
Inversion in $\mathbb{F}_p$	50
Final exponentiation in $\mathbb{F}_{p^2}$	202
Tate pairing	752
1024-RSA decryption	242

4.1. Timing result

The porting of Tate pairing algorithm has been developed entirely in software without using any dedicated hardware but making use of the specific instruction set to carefully implement the optimizations discussed in the previous sections.

In Table 4, the timing of the Tate pairing computation for a finite field of size $\approx 2^{512}$ is compared with an optimized implementation of the 1024-RSA decryption function (without CRT), specifically designed for the ST22 platform (STMicroelectronics, 2005a) that suppose the same security level.

The execution time of the pairing primitives compared with a 1024-RSA decryption is three times longer, but the absolute value of 752 ms is still significant and totally acceptable for real applications. We outline that the implementation of the Tate pairing over fields of characteristic $p > 3$ shares with RSA the same low level arithmetic instructions. Therefore the cryptographic functions of a

[1] Proprietary Instruction Set Architecture details are not publicly available.

Table 5
Execution time comparison on different Smart-Card platform

Platform	Pairing algorithm	Time [ms]
Philips HiPerSmart™	BKLS	470 (@ 21 MHz) 290 estimated (@ 36 MHz)
Philips HiPerSmart™	Ate	590 (@ 21 MHz) 380 estimated (@ 36 MHz)
SmartJ ST22	BKLS	752 (@ 33 MHz)

Smart-Card can be easily extended to include pairing-based features.

Most recent related works about implementation of pairings on constrained platforms can be found in Scott et al. (2006), where Scott et al. report the timings relative to various pairing algorithms on the Philips HiPerSmart™ Smart-Card with a MIPS-32 processor (five stage pipeline 2KB instruction cache, 256KB flash memory, 16KB DRAM). For the same security level considered in Table 4 the authors consider the Tate pairing on a non-supersingular curve following the description of the BKLS algorithm analyzed in Scott (2005) over $\mathbb{F}_p$ and the Ate pairing on non-supersingular curve as discussed in Hess et al. (2006) over $\mathbb{F}_p$. The reported timings compares favorably with the implementation described in the current paper and are shown in Table 5.

Other previously reported implementations can be only referred as an announcement by Gemplus (2005) which was quite contemporary with the presentation of the results discussed in this paper at InfoSecurity 2005 Exhibition (STMicroelectronics, 2005b).

5. Concluding remarks

The Tate pairing is one of the basic building blocks for Pairing Encryption primitives. We have described a methodology to choose the optimal configuration of the Tate pairing parameters (p, l, c) using supersingular elliptic curves over $\mathbb{F}_p$. Besides, a customized hybrid coordinate system for the elliptic curve points has been defined to merge the double-and-add scheme with the computation of the pairing value, minimizing the overall number of operations in the ground field. To speed up modular reduction, we re-formulated the classical division algorithm to obtain an effective implementation tailored for the specific modulus, the size of the operands and the memory hierarchy with no data or instruction cache. The execution time of the proposed implementation on the ST22 Smart-Card proves that the computation of the Tate pairing primitive in embedded devices achieves significant and acceptable performances for real applications.

References

Adi Shamir, 1984. Identity-Based Cryptosystems and Signature Schemes. In: CRYPTO, pp. 47–53.

Balasubramanian, R., Koblitz, Neal, 1998. The improbability that an elliptic curve has subexponential discrete log problem under the Menezes–Okamoto–Vanstone algorithm. Journal of Cryptology 11 (2), 141–145.

Barreto, Paulo S.L., 2007. The Pairing-based Crypto Lounge. <http://paginas.terra.com.br/informatica/paulobarreto/pblounge.html>.

Barreto, Paulo S.L.M., Kim, Hae Yong, Lynn, Ben, Scott, Michael, 2002a. Efficient algorithms for pairing-based cryptosystems. In: CRYPTO '02: Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology. Springer-Verlag, London, UK, pp. 354–368.

Barreto, Paulo S.L.M., Lynn, Ben, Scott, Michael, 2002b. Constructing elliptic curves with prescribed embedding degrees. In: SCN. In: Cimato, Stelvio, Galdi, Clemente, Persiano, Giuseppe (Eds.), Lecture Notes in Computer Science, vol. 2576. Springer, pp. 257–267.

Barreto, P., Galbraith, S., hEigeartaigh, C., Scott, M., 2004. Efficient pairing computation on supersingular abelian varieties. Cryptology ePrint Archive, Report 2004/375. <http://eprint.iacr.org/2004/375.ps>.

Blake, Ian F., Seroussi, Gadiel, Smart, Nigel, 2006. Advances in Elliptic Curve Cryptography. Cambridge University Press.

Boneh, Dan, Franklin, Matthew K., 2001. Identity-based encryption from the weil pairing. In: CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology. Springer-Verlag, London, UK, pp. 213–229.

Boneh, Dan, Lynn, Ben, Shacham, Hovav, 2001. Short signatures from the weil pairing. In: ASIACRYPT '01: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security. Lecture Notes in Computer Science. Springer-Verlag, London, UK, pp. 514–532.

Boneh, Dan, Boyen, Xavier, Shacham, Hovav, 2004. Short group signatures. In: Advances in Cryptology—CRYPTO 2004. Lecture Notes in Computer Science, vol. 3152. Springer-Verlag, Berlin, pp. 41–55. Available at <http://www.cs.stanford.edu/xb/crypto04a/>.

Chudnovsky, D.V., Chudnovsky, G.V., 1986. Sequences of numbers generated by addition in formal groups and new primality and factorization tests. Advances in Applied Mathematics 7 (4), 385–434.

Cocks, Clifford, 2001. An identity based encryption scheme based on quadratic residues. In: IMA Int. Conf. In: Honary, Bahram (Ed.), Lecture Notes in Computer Science, vol. 2260. Springer, pp. 360–363.

Cohen, Henri, Miyaji, Atsuko, Ono, Takatoshi, 1998. Efficient elliptic curve exponentiation using mixed coordinates. In: ASIACRYPT '98: Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security. Springer-Verlag, London, UK, pp. 51–65.

Dutta, R., Barua, R., Sarkar, P., 2004. Pairing-based cryptographic protocols: a survey. Cryptology ePrint Archive, Report 2005/64. <http://eprint.iacr.org/2004/64.pdf>.

Duursma, Iwan M., Lee, Hyang-Sook, 2003. Tate pairing implementation for hyperelliptic curves $y^2 = x^p - x + d$. In: ASIACRYPT. In: Laih, Chi-Sung (Ed.), Lecture Notes in Computer Science, vol. 2894. Springer, pp. 111–123.

Frey, Gerhard, Müller, Michael, Rück, Hans-Georg, 1999. The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems. IEEE Transactions on Information Theory 45 (5), 1717–1719.

Galbraith, Steven D., Harrison, Keith, Soldera, David, 2002. Implementing the tate pairing. In: ANTS-V: Proceedings of the 5th International Symposium on Algorithmic Number Theory. Springer-Verlag, London, UK, pp. 324–337.

Galbraith, S., McKee, J., Valenca, P., 2004. Ordinary abelian varieties having small embedding degree. Cryptology ePrint Archive, Report 2004/365. <http://eprint.iacr.org/2004/365.pdf>.

Gemplus, 2005. ID based cryptography and Smartcards. <http://www.gemplus.com/smart/rd/pubblications/pdf/Joy05iden.pdf>.

Gura, Nils, Patel, Arun, Wander, Arvinderpal, Eberle, Hans, Shantz, Sheueling Chang, 2004. Comparing elliptic curve cryptography and RSA on 8-bit CPUs. In: CHES. In: Joye, Marc, Quisquater, Jean-Jacques (Eds.), Lecture Notes in Computer Science, vol. 3156. Springer, pp. 119–132.

Hess, Florian, Smart, Nigel P., Vercauteren, Frederick, 2006. The eta pairing revisited. IEEE Transactions on Information Theory 52 (10), 4595–4602.

International Association for Cryptologic Research, 2007. Cryptology ePrint Archive. <http://eprint.iacr.org>.

Joux, Antoine, 2000. A one round protocol for tripartite Diffie–Hellman. In: ANTS-IV: Proceedings of the 4th International Symposium on Algorithmic Number Theory. Springer-Verlag, London, UK, pp. 385–394.

Karatsuba, A., Ofman, Yu, 1962. Multiplication of many-digital numbers by automatic computers. Doklady Akademii Nauk SSSR, 145, 293–294. Translation in Physics-Doklady, 7, 595–596, 1963.

Koblitz, Neal, Menezes, Alfred, 2005. Pairing-based cryptography at high security levels. In: IMA Int. Conf.. In: Smart, Nigel P. (Ed.), Lecture Notes in Computer Science, vol. 3796. Springer, pp. 13–36.

Menezes, Alfred, Okamoto, Tatsuaki, Vanstone, Scott A., 1993. Reducing elliptic curve logarithms to logarithms in a finite field. IEEE Transactions on Information Theory 39 (5), 1639–1646.

Menezes, Alfred J., Vanstone, Scott A., Van Oorschot, Paul C., 1996. Handbook of Applied Cryptography. CRC Press Inc., Boca Raton, FL.

Mont, Marco Casassa, Bramhall, Pete, Harrison, Keith, 2003. A flexible role-based secure messaging service: exploiting ibe technology for privacy in health care. In: DEXA '03: Proceedings of the 14th International Workshop on Database and Expert Systems Applications. IEEE Computer Society, Washington, DC, USA, p. 432.

Montgomery, Peter L., 1985. Modular multiplication without trial division. Mathematics of Computation 44, 519–521.

Sakai, R., Ohgishi, K., Kasahara. M., 2000. Cryptosystems based on pairing. In: Symposium on Cryptography and Information Security (SCIS2000).

Scott, Michael, 2005. Computing the tate pairing. In: CT-RSA. In: Menezes, Alfred (Ed.), Lecture Notes in Computer Science, vol. 3376. Springer, pp. 293–304.

Scott, Michael, Barreto, Paulo S.L.M., 2006. Generating more MNT elliptic curves. Designs Codes and Cryptography 38 (2), 209–217.

Scott, Michael, Costigan, Neil, Abdulwahab, Wesam, 2006. Implementing cryptographic pairings on smartcards. In: CHES. In: Goubin, Louis, Matsui, Mitsuro (Eds.), Lecture Notes in Computer Science, vol. 4249. Springer, pp. 134–147.

Silverman, Joseph H., 1994. The arithmetic of elliptic curves, xii ed. Graduate Texts in Mathematics, vol. 106 Springer-Verlang.

Solinas, Jerome A., 1999. Generalized Mersenne Numbers. Techical Report CORR 99-39, Centre for Applied Cryptographic Research, University of Waterloo.

STMicroelectronics, 2005. Data Brief ST22 Smart J Platform Smartcard ICs, February. <http://www.st.com/stonline/products/families/smartcard/>.

STMicroelectronics, 2005. Encryption and smart card technology leaders develop identifier-based encryption for portable formats, April. <http://www.st.com/stonline/press/news/year2005/t1621m.htm>.

Verheul, E., 2001. Evidence that XTR is more secure than supersingular elliptic curve cryptosystems. Advances in Cryptology – Eurocrypt 2001, LNCS 2045, 195–210.

Victor, Miller, 1986. Short programs for functions on curve. Unpublished manuscript. Avaliable at <http://crypto.stanford.edu/miller/miller.pdf>.

Zhang, Fangguo, Kim, Kwangjo, 2002. Id-based blind signature and ring signature from pairings. In: ASIACRYPT '02: Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security. Springer, London, UK, pp. 533–547.