Pairing in Cryptography: an Arithmetic Point of View

Jean-Claude Bajard, Nadia El Mrabet

Pairing in cryptography: an arithmetic point of view

J.C. Bajard, N. El Mrabet

ARITH-LIRMM, CNRS Université Montpellier2, France

ABSTRACT

The pairing is a mathematical notion wich appeared in cryptography during the 80'. At the beginning, it was used to build attacks on cryptosystems, transferring the discrete logarithm problem on elliptic curves, to a discrete logarithm problem on finite fields, the first was the MOV[36] attack in 1993. Now, pairings are used to construct some cryptographic protocols: Diffie Hellman tripartite, identity based encryption, or short signature. The main two pairings usually used are the Tate and Weil pairings. They use distortions and rationnal functions, and their complexities depends of the curve and the field involved.

This study deals with two particular papers: one due to N. Koblitz and A. Menezes[27] published in 2005, and a second one written by R Granger, D. Page and N. Smart[24] in 2006. These two papers compare Tate and Weil pairings, but they differ in their conclusions. We consider the different arithmetic tricks used, trying to precise each point, in a way to avoid any ambiguity. Thus, the arithmetics proposed take into account the features of the fields and the curves used. We clarify the complexity of the possible implementations. We compare the different approaches, in order to clarify the conclusions of the previous papers.

Keywords: Pairing, arithmetic operators

1. INTRODUCTION

Cryptographic protocols are divided in two main classes, symmetric systems where keys are secret, and asymmetric approaches with public keys. The security of this second category is based on problems known to be difficult to solve, we speak about proved cryptography. Historically, in 1976, Diffie-Hellman protocol[15] was one of the first crypto-systems based on the discrete logarithm problem. Then, with the introduction of the elliptic curve in cryptography which was promoted by V. Miller[33] and N. Koblitz,[26] a large spectrum of crypto-systems appeared. Pairings are bilinear maps which allow to transform an approach on abelean curves, as elliptic ones, to a problem on finite fields. A first use of such maps concerns cryptanalysis, and was proposed in 1994 by G. Frey and H.G. Rück[21] whose link pairings to the discrete logarithmic problem on curves.[19,21] The construction of the pairings is based on the algorithm proposed in 1986 by Victor Miller.[32,34]

In 2000, A. Joux[25] had proposed a tripartite Diffie-Hellmann keys exchange using pairing. That was the beginning of a blossoming literature on the subject. In 2003, D. Boneh and M. Franklin broke a challenge given by Shamir[40] in 1984, creating an identity-based encryption scheme[10] based on pairings. Since the literature is very rich on this subject,[39] some related conferences are born, from this interest, as Pairings.[38]

With the birth of this new domain of investigation in cryptography, the problem of implementing these protocols occurs. This point is very relevant to the interest of pairings, the cost and the performances of the implementation make a cryptosystem available. Some good studies on pairings implementation are given by P. Barreto et al,[4,6] we can also refer to some books.[17,22]

We focus our study on the complexity of the implementations proposed in two papers: a first one using friendly fields[27] for an embedded degree of the form 2^i3^j, and a second one based on a cyclotomic representation of the fields.[24]

Organization of the paper is the following: first we introduce the pairings and the Miller algorithm, then we present the different arithmetic tricks used by N. Koblitz and A. Menezes,[27] and by R. Granger, D. Page and N. Smart,[24] we end by a discussion on this two papers, diving some ideas for improvements.

Further author information: JC Bajard: E-mail: bajard@lirmm.fr.

2. SHORT INTRODUCTION TO PAIRINGS

Referring to a recent book,[14] pairings is an old mathematic notion.[16] The main idea consists in a bilinear map from a product of two groups due to an elliptic curve, to a group over a finite field. This bilinear map is called pairing. In mathematic,[37] this notion is extended to abelian varieties, but for our purpose we will restrain our study to elliptic curves pairings used in cryptography. For a general introduction we can refer to some books[16,22]

Summarizing, we define a pairing as following: we consider, $n \in \mathbb{N}^*$, G_1 and G_2 two additive abelean groups of cardinal n and G_3 a cyclic (multiplicative) group of cardinal n.

DEFINITION 2.1. *A pairing is a function $e : G_1 \times G_2 \to G_3$ which verifies the following properties:*

- *Bilinearity:* $\forall P, P' \in G_1, \forall Q, Q' \in G_2$

$$\begin{aligned} e(P + P', Q) &= e(P, Q).e(P', Q) \\ e(P, Q + Q') &= e(P, Q).e(P, Q') \\ e(P, iQ) &= e(P, Q)^i \end{aligned}$$

- *Non-degeneracy:*

$$\begin{aligned} &\forall P \in G_1 - \{0\}, \quad \exists Q \in G_2, \quad \textit{such that:} \quad e(P, Q) \neq 1 \\ &\forall Q \in G_2 - \{0\}, \quad \exists P \in G_1, \quad \textit{such that:} \quad e(P, Q) \neq 1 \end{aligned}$$

For our purpose, we only consider parings defined on elliptic curves in a finite field $\mathbb{F}_p$, more precisely $G_1 \subset E(\mathbb{F}_p)$, $G_2 \subset E(\mathbb{F}_{p^k})$ and $G_3 \subset \mathbb{F}_{p^k}$ with E an elliptic curve defined by the equation $y^2 = x^3 + ax + b$ (in fact we can chose $a = -3$),[11] and where k is the smallest integer such that n divides $p^k - 1$, k is called the embedded degree of the curve.

The group G_1 is a n torsion subgroup of $E(\mathbb{F}_p)$: $E[n] = \{P \in E(\mathbb{F}_p),\ [n]P = P_\infty\}$ (subset of points of order n, with P_∞ the neutral element of the group). Due to the non-degeneracy, G_2 must be different to G_1, thus we chose it as the n torsion subgroup of $E(\overline{\mathbb{F}_p})$, where $\overline{\mathbb{F}_p}$ is the algebraic closure of $\overline{\mathbb{F}_p}$. In fact, G_2 is chosen as the n torsion subgroup of $E(\mathbb{F}_{p^k})$. It is a subgroup of points belonging to $E(\mathbb{F}_{p^k})$, of order n . Then, G_3 corresponds to U_n the group of the n^{th}-roots of the unity in $\mathbb{F}_{p^k}$: $U_n = \{\xi \in \mathbb{F}_{p^k},\ \text{such that:}\ \xi^n = 1\}$.

The link to the discrete logarithm is due to the bilinearity of e: $e(aP, bQ) = e(P, Q)^{ab}$, which is used in the cryptographic protocols. A. Menezes gives[30] a good introduction to pairings applied to cryptography, another tutorial[2] is given by P. Barreto, one of the main authors on this domain. We can also refer to K.G. Paterson.[39]

Pairings implementations are linked to an algorithm due to V. Miller.[32,34] The two most famous pairings, Weil pairing and Tate pairing are based on this algorithm. Thus, we present this algorithm in his cryptographic application, and then we give the definitions of these two pairings.

2.1. Miller algorithm

The description given here, is directly inspired from lectures given by T. Lange[28] and E. Thomé,[42] and a book chapter written by S. Duquene and G. Frey.[17]

The main idea is based on the notion of divisors,[12,14] to simplify, we summarize this construction by giving only the elements needed for the evaluation. The goal of Miller algorithm is to construct a rational function F_P associated to the point P (P is a generator of G_1), and to evaluate it at a point Q (in fact a divisor of this point). The function F_P is such that P is a generator of order n and P is a root of degree n of this function (for the divisor point of view, $D_{F_P} = n(P) - n(P_\infty)$).

V. Miller proposed in 1986[32,34] an algorithm for the construction of a such function and its evaluation. For that, he builds F_P with an iterative process using the double & add construction of nP. At each step, we consider for the numerator (roots), a straight line which passes by the points of the construction of nP (defined by two different points or a tangent), and for the denominator (pole), a vertical one passing by the sum of these points.

Algorithm 1: Miller(P, Q, n)

Data: $n = (n_l \dots n_0)$(radix 2 representation), $P \in G_1(\subset E(\mathbb{F}_p))$ et $Q \in G_2(\subset E(\mathbb{F}_{p^k}))$;
Result: $F_P(Q) \in G_3(\subset \mathbb{F}^*_{p^k})$;
$T \leftarrow P$;
$f_1 \leftarrow 1$;
$f_2 \leftarrow 1$;
for $i = l - 1$ **to** 0 **do**
1 $T \leftarrow [2]T$;
$f_1 \longleftarrow {f_1}^2 \times h_1(Q)$;
$f_2 \longleftarrow {f_2}^2 \times h_2(Q)$ (where $Div(\frac{h_1}{h_2}) = 2(T) - ([2]T) - P_\infty$);
2 **if** $n_i = 1$ **then**
$T \leftarrow T \oplus P$;
$f_1 \longleftarrow f_1 \times h_1(Q)$;
$f_2 \longleftarrow f_2 \times h_2(Q)$ (where $Div(\frac{h_1}{h_2}) = (T) + D_P - ((T) \oplus D_P) - P_\infty$);
end
end
return $\frac{f_1}{f_2}$

In the part 1 of algorithm 1, the function h_1 is due to the equation $h_1(X, Y) = 0$ of the tangent to the curve E at the point T (that corresponds to the doubling $2T$), then h_2 is associated to the straight line define by the point $(2T)$ and the point at infinity P_∞. The part 2 is done only if the digit of n considered is equal to one. In this case, h_1 corresponds to the straight line defined by T and P, and h_2 comes from the one due to the point $(T + P)$ and P_∞.

2.2. Arithmetic aspects

P is a point of $E(\mathbb{F}_p)$ considered in affine coordinates (X_P, Y_P) (or in Jacobean with $Z_p = 1$). We remark that P is also a point of $E(\mathbb{F}_{p^k})$ considering that $\mathbb{F}_p \subset \mathbb{F}_{p^k}$. The point Q belonging to $E(\mathbb{F}_{p^k})$, is also given in affine coordinates (X_Q, Y_Q). T is a point of $E(\mathbb{F}_p)$ considered in Jacobean projective coordinates (X_T, Y_T, Z_T) (with the corresponding affine coordinates $(X_T/Z_T^2, Y_T/Z_T^3)$) to avoid inversion in $\mathbb{F}_{p^k}$. The values f_1 and f_2 belong to $\mathbb{F}_{p^k}$. And, $h_1(Q)$ and $h_2(Q)$ are also elements of $\mathbb{F}_{p^k}$.

The choice of n is consequent, thus it is generally selected to have few ones in its binary representation (thus the addition, part 2 of algorithm 1, can be neglected). Subsequently, we focus our attention one the doubling part of the algorithm 1. The equation of the curve E is of the form $y^2 = x^3 - 3xz^4 + bz^6$ (we can consider that $a = -3$),[11] thus for $T(X_T, Y_T, Z_T)$, the point $2T(X_{2T}, Y_{2T}, Z_{2T})$ is given by:[18]

$$C = 2Y_T^2, \quad D = Z_T^2, \quad A = 4X_TY_T^2 = 2X_TC, \quad B = (3X_T^2 - 3Z_T^4) = 3(X_T - D)(X_T + D) \tag{1}$$

$$X_{2T} = B^2 - 2A, \quad Y_{2T} = B(A - X_{2T}) - 2C^2, \quad Z_{2T} = 2Y_TZ_T. \tag{2}$$

In this case, h_1 et h_2, with $Q = (X_Q, Y_Q) \in E(\mathbb{F}_{p^k})$, are obtained with:

$$D' = Z_{2T}D, \quad D_2 = Z_{2T}^2, \quad h_1 = D_2[D'Y_Q - C - B(X_QD - X_T)], \quad h_2 = (D')(D_2X_Q - X_{2T}). \tag{3}$$

We remark that some terms have to be computed only one time, for example: Y_T^2, Z_T^2, $4X_TY_T^2$, $(3X_T^2 + aZ_T^4)$. Thus the number of operations is reduced to, for the doubling, $4Squ+4Mul$ and for the function h evaluation $1Squ + 7Mul$.

Now, if we consider the fields on which are defined the points P and Q, the complexity of this operation in algorithm 1 requires $5Squ + (13 + 2k)Mul$ in F_p taking into account that the values obtained from the doubling part, belong to F_p and we only have to multiply two elements (the coordinates of Q) from $\mathbb{F}_{p^k}$ by such values. For a complete step of the doubling in Miller algorithm we have to square f_1 and f_2, and to multiply these values by h_1 and h_2, thus, we have to perform $2Squ+2Mul$ in $\mathbb{F}_{p^k}$. We will see how to reduce this complexity by using twisted curves in section 3.1.

Now for obtaining a pairing, we must insure the non-degeneracy of the function and that the images in $\mathbb{F}_{p^k}$ are order n elements of this field. Thus, we present two well known pairings: the Weil pairing and the Tate paring.

As we want to make this paper easy to read, we simplify the writing, avoiding the use of the divisor terminology used in mathematics. Thus, we note $F_P(D_Q)$ the result of Miller algorithm for P and Q as arguments.

2.3. Definition of Weil pairing

The Weil pairing is define for $P \in G_1$, and $Q \in G_2$. We note it e_W, such that:

$$\begin{aligned} G_1 \times G_2 &\mapsto G_3 \\ (P, Q) &\mapsto e_W(P, Q) = \frac{F_P(D_Q)}{F_Q(D_P)} \end{aligned}$$

In this pairing, Miller algorithm is applied two times. But we remark that, if for $F_P(D_Q)$ we use the classical Miller (or Lite Miller), the evaluation of $F_Q(D_P)$ deals with $Q \in G_2$ (Full Miller), thus its complexity is bigger due to the evaluation of nQ and the fact that $T \in G_2$. The evaluation ends with an inversion in G_3 (which is a subgroup of a finite field) and a product by this inverse.

2.4. Definition of Tate pairing

By the same way, we define the Tate pairing, noted e_T:

$$\begin{aligned} G_1 \times G_2 &\mapsto G_3 \\ (P, Q) &\mapsto e_T(P, Q) = F_P(D_Q)^{\frac{p^k-1}{n}} \end{aligned}$$

Here, $F_P(D_Q)$ is evaluated with the Lite Miller algorithm. Then, this value is powered to $\frac{p^k-1}{n}$ by a classical exponentiation algorithm which can use a sliding window approach. This last operation can be improved by different ways, as we will see in section 3.3.

2.5. Comparison of Weil and Tate Pairings

These two pairings are often compared. The difference is due to the Full Miller completed by an inversion and a product in the Weil paring, and the final exponentiation in the Tate pairing. At the first time, Weil seems more expensive than Tate, that is admitted by the community.

But in 2005, N. Koblitz and A. Menezes[27] made the comparaison between the cost of the Weil and Tate pairings, and for high security levels, they concluded that Weil becomes more efficient than Tate, this is due to the expensive cost the final exponentiation. Then, in 2006, R. Granger, D. Page and N. Smart[24] replied to Koblitz and Menezes, improving the arithmetic in extension field, and in particular the final exponentiation, they concluded that for current security levels, Tate is always more efficient than Weil.

In the next section, we present the different tricks used for improving the calculus. We verified the formula of the complexity given in the previous papers, using the same criteria, and we extend the evaluation to upper security levels to find when Weil becomes better.

3. REMARKS ON THE ARITHMETIC IMPLEMENTATION

3.1. Twisted curves: a way to reduce the calculus and to avoid the denominator evaluation in Miller algorithm

For decreasing the complexity of the evaluation, we can consider the twisted curve $\tilde{E}(\mathbb{F}_{p^{k/2}})$ of $E(\mathbb{F}_{p^{k/2}})$.[7]

We consider $\nu \in \mathbb{F}_{p^{k/2}}$ a non-quadratic residue (with $\sqrt{\nu} \in \mathbb{F}_{p^k}$), we can define $\tilde{E}(\mathbb{F}_{p^{k/2}})$ a twisted curve of $E(\mathbb{F}_{p^{k/2}})$ of equation: $\nu y = x^3 - 3x + b$. There exists an isomorphism ψ from a subgroup of $\tilde{E}(\mathbb{F}_{p^{k/2}})$ to a subgroup of $E(\mathbb{F}_{p^k})$ which maps $Q'(X,Y) \in \tilde{E}(\mathbb{F}_{p^{k/2}})$ to $Q(X, \sqrt{\nu}Y) \in E(\mathbb{F}_{p^k})$ (the probability that Q belongs to the group generated by $P \in E(\mathbb{F}_p)$ is weak[7]). Hence, the evaluations with the coordinates of Q are made in $\mathbb{F}_{p^{k/2}}$, that reduces the cost of the Miller algorithm.

Another interesting fact is, that, for the Tate pairing, we don't have to take care of the evaluation of f_2 and thus of h_2.[7] It is due to $h_2 \in \mathbb{F}_{p^{k/2}}$, so $f_2 \in \mathbb{F}_{p^{k/2}}$, and to n which divides $p^k - 1$, but it does not divide $p^{k/2} - 1$ (definition of the embedded degree), so $\frac{p^k-1}{n}$ is a multiple of $p^{k/2} - 1$, then, as for all element γ of $\mathbb{F}_{p^{k/2}}$, $\gamma^{p^{k/2}-1} \equiv 1$, we have $f_2^{\frac{p^k-1}{n}} \equiv 1$. We can do the same remark about the square D_2 which is an element of $\mathbb{F}_{p^{k/2}}$ and can be neglected in the formula (3).

This remark can be applied to Weil pairing,[27] where, for a cryptographic application, we can replace the pairing value by a power of it, such that the exponent is not divisible by n. Hence, if we consider $e_w(P,Q)^{p^{k/2}-1}$ as result, we don't have to take care of the denominator part (we don't compute f_2 and the product by D_2).

Hence, Lite Miller is reduced, for the doubling step, to $4Squ+4Mul+(3+k)Mul$ in $\mathbb{F}_p$ and $1Squ+1Mul$ in $\mathbb{F}_{p^k}$. This remark is also available for the Full Miller, whose complexity is $4Squ + 4Mul + 2Mul$ in $\mathbb{F}_{p^{k/2}}$, $kMul$ in $\mathbb{F}_p$ (remember that $P \in E(\mathbb{F}_p)$), and $1Squ + 1Mul$ in $\mathbb{F}_{p^k}$.

3.2. Friendly Fields

When we consider $\mathbb{F}_{p^k}$, a first remark occurs if k is composite. In this case, if $k = lm$ then we can consider $\mathbb{F}_{p^k}$ as a l extension of $\mathbb{F}_{p^m}$. In other words, if we take into account the decomposition of k as a product of primes, then we obtain a multi-decomposition of the extension, this can permit to decrease the number of operations.[20]

A.Menezes and N.Koblitz[27] propose to use friendly pairing fields for improving the computations in extensions of the finite fields, for the pairing based cryptography. These fields improve the computation in the extension, they simplify the analysis of the cost of multiplication used in pairings.

DEFINITION 3.1. *A friendly pairing field is an extension $\mathbb{F}_{p^k}$ of a finite field F_p with the following properties:*

- *p the characteristic of the finite field verifies $p \equiv 1 \ mod(12)$,*
- *k the embedded degree of the curve is such that: $k = 2^i 3^j$.*

THEOREM 3.2. *Let $\beta \in \mathbb{F}_p$ be neither a square nor a cube in $\mathbb{F}_p$, and $\mathbb{F}_{p^k}$ be a friendly pairing field, then the polynomial $X^k - \beta$ is irreducible on $\mathbb{F}_p$.*

Hence, $\mathbb{F}_{p^k} = \mathbb{F}_p[X]/(X^k - \beta)$, and it can be considered as a tower of extensions of degree 2 (quadratic extension) and 3 (cubic extension). For the construction, we consider the square or cubic root of β and then the square or cubic root of the result. We can chose β as a small value of $\mathbb{F}_p$, then the multiplications by β can be reduced to few additions, and its cost can be neglected.

Example: Construction for for $k = 2^3 3^1$:

$$\begin{aligned} \mathbb{F}_p \xrightarrow{3} K &= \mathbb{F}_p[T]/(T^3 - \beta) \\ K \xrightarrow{2} L &= K[U]/(U^2 - T) \\ L \xrightarrow{2} M &= L[V]/(V^2 - U) \\ M \xrightarrow{2} N &= M[W]/(W^2 - V) \end{aligned}$$

We can construct the arithmetic in $\mathbb{F}_{p^k}$, step by step in smaller extensions. For a product in the extension $\mathbb{F}_{p^k}$, Karatsuba and Tom-Cook methods reduce the numbers of operations in $\mathbb{F}_p$ using this decomposition. Thus, one multiplication of two elements of $\mathbb{F}_{p^k}$ is done with $\rho(k)Mul$ in $\mathbb{F}_p$, where $\rho(2^i 3^j) = 3^i 5^j$.

3.3. Improving the final exponentiation

The Tate pairing is composed of two steps, first the computation of the Miller Lite number, and then, the exponentiation of this result. The exponentiation is done in $\mathbb{F}_{p^k}$ and that could be expensive. Koblitz and Menezes[27] split the exponentiation in two parts. The first corresponds to a computation using properties of the twisted curve and the friendly fields, which is not too expensive, and the second corresponds to a reduced exponentiation in $\mathbb{F}_{p^k}$.

Thus, we want to compute: $\omega^{\frac{q^k-1}{n}}$, where $\omega \in \mathbb{F}_{p^k}$. For that, they suggest to split this operation in two parts : $\omega^{\frac{p^k-1}{n}} = \left(\omega^{\frac{p^k-1}{\Phi_k(p)}}\right)^{\frac{\Phi_k(p)}{n}}$, where ϕ_k is the k^{th} cyclotomic polynom evaluated in p. The exponent can be split like this, because $\phi_k(p)$ divides $(p^k - 1)$ and n divides $\phi_k(p)$.

First part: We first remark that : if $k = 2^i 3^j$, then $\Phi_k(p) = p^{k/3} + p^{k/6} + 1$ and $\frac{p^k-1}{\Phi_k(p)} = (p^{k/2} - 1)(p^{k/6} + 1)$. Now, if we assume that we use the representation due to the twisted curve (see section 3.1), then the result in $\mathbb{F}_{p^k}$ of the Miller algorithm is of the form $(X + Y\sqrt{\nu})$ with $X, Y \in \mathbb{F}_{p^{k/2}}$.

Thus, we get $(X+Y\sqrt{\nu})^{p^{k/2}-1} = (X+Y\sqrt{\nu}^{(p^{k/2}-1)})$, and we just have to evaluate $(X+Y\sqrt{\nu}^{(p^{k/2}-1)})^{p^{k/6}+1}$ which corresponds to Frobenius operations. Hence, for $\omega \in \mathbb{F}_{p^k}$ and $\omega = \sum_{i=0}^{k-1} a_i \xi^i$, where $a_i \in \mathbb{F}_p$ and ξ is a root of $X^k - \beta$, the polynom used to build the extension $\mathbb{F}_{p^k}$, we have $\omega^p = \sum_{i=0}^{k-1} a_i \xi^{ip}$.

We remark that: $\xi^p = \xi^{[k(p-1)/k]+1} = \beta^{(p-1)/k}\xi = \beta^{(p-1)/k}\xi$. Denoting $\theta = \beta^{(p-1)/k}$, we obtain $\xi^p = \theta.\xi$ and $\xi^{p^i} = \theta^i.\xi$. Thus, the cost of this part of the exponentiation can be neglected.

Second part: For improving the exponentiation to $\frac{\Phi_k(p)}{n}$, we can used the Lucas sequences.[8] The cost of the exponentation using this method is one multiplication and one square in the intermediate field $\mathbb{F}_{p^{k/2}}$ for each bit of the exponent. Another approach deals with sliding windows, in this case[23] squaring can be done in the cyclotomic subgroup. Here too, the complexity is linearly dependent of the number of bits of n.

Let b_n be the minimun number of bits of n and b_{p^k} the minimun number of bits of p^k. The exponent is then, composed of $(\frac{\varphi(k)}{k} b_{p^k} - b_n)$ bits (i.e. $\varphi(k)$ is the degree of $\Phi_k(p)$, and $b_p \sim b_{p^k}/k$).

Hence, the number of multiplications and squares required for this part, depends of: $\frac{\varphi(k)}{k} b_{p^k} - b_n = (\tau_k \gamma - 1) b_n$, where $\tau_k = \frac{\varphi(k)}{k} = \begin{cases} 1/2 & \text{si } k = 2^i,\ i \geqslant 1 \\ 1/3 & \text{si } k = 2^i 3^j,\ i,\ j \geqslant 1 \end{cases}$. The number γ which is the ratio $\frac{b_{p^k}}{b_n}$, is related to the security levels in table 1, and gives an idea of the degree of complexity.

security level (bits)	80	128	192	256	384
b_n nb min of bits of n	60	256	384	512	768
b_{p^k} nb min of bits of p^k	1024	3072	8192	15 360	26 880
$\gamma = \frac{b_{p^k}}{b_n}$	6,4	12	21,33	30	35

Table 1. Security level.

3.4. Remarks about inversions

We just have to remark that n, the order of the subgroup G_3, is prime and divides $p^{k/2}+1$ (we recall that k is the smallest value such that n divides p^k-1 and k is even). Then, we remark that if $\alpha \in G_3$, $\alpha^{-1} = \alpha^{p^{k/2}}$, because : $\alpha \times \alpha^{p^{k/2}} = \alpha^{p^{k/2}+1} = \alpha^{n.\frac{p^{k/2}+1}{n}} = 1$

Hence, the final inversion in Weil, is reduced to a composition of Frobenius, as seen in the first part of section 3.3.

3.5. Cyclotomic subgroup and squaring

A. Lenstra and M. Stam[29] introduced a very efficient and original way for improving the evaluation of the square in a cyclotomic subgroup. A cyclotomic subgroup, $G_{\phi_k(p)}$ is a subgroup of order $\phi_k(p)$, where ϕ_k is the k^{th} cyclotomic polynomial evaluated in p. This method is interesting for the squaring in the subgroup of embedded degree 6 or a multiple of 6, the other embedded degrees have not yet been studied.

They consider $\phi_k(X) = X^{k/3} - X^{k/6} + 1$, and they define $\alpha \in G_{\phi_k(p)}$ by $\alpha = \sum_{i=1}^{k} a_i \xi^i$, where the a_i are used as variables in $\mathbb{F}_p$. Symbolically, we can compute $\alpha^{p^{k/3}}$, $\alpha^{p^{k/6}}$, and the equation $\alpha.\alpha^{p^{k/3}} - \alpha^{p^{k/6}} = \sum_{i=1}^{k} v_i \xi^i$ gives the v_i in function of the a_i and β.
In fact, the subgroup $G_{\phi_k(p)}$ is the variety constituted with the α such that $\forall i,\ v_i = 0$. Symbolically, we can write that $\alpha^2 = \alpha^2 + b.\Gamma.^t v$, with $b = (1, \xi, \xi^2, ..., \xi^{k-1})$, and Γ a given matrix which allows to simplify the expression of the square. If we note $\alpha^2 = \sum_{i=1}^{k} s_i \xi^i$, then we have $\sum_{i=1}^{k} s_i \xi^i = (\sum_{i=1}^{k} a_i \xi^i)^2 + b.\Gamma.^t v$.

Cost of the square obtained by R Granger, D Page anf N. Smart[24] is:

k	Cost of a square ($\mathbb{F}_p$op.)
6	$6Squ + 3Mul$
12	$18Mul + 12Squ$
24	$84Mul + 24Squ$

They remark that for $k = 6$ and $p \equiv 2 \pmod 9$, the squaring can be improved, using a representation of $\mathbb{F}_{p^6}$ defined by a root of $G(X) = X^6 + X^3 + 1$. In this case, the formula are simpler, and a squaring needs only $6Mul$ in $\mathbb{F}_p$.

3.6. Weil or Tate

We have seen (in section 3.1), that the complexity of Miller algorithm can be summarize by:

Lite Miller C_{Lite}	$(4Squ_{\mathbb{F}_p} + (7+k)Mul_{\mathbb{F}_p} + 1Squ_{\mathbb{F}_{p^k}} + 1Mul_{\mathbb{F}_{p^k}})\log_2(n)$
Full Miller C_{Full}	$(kMul_{\mathbb{F}_p} + 4Squ_{\mathbb{F}_{p^{k/2}}} + 6Mul_{\mathbb{F}_{p^{k/2}}} + 1Squ_{\mathbb{F}_{p^k}} + 1Mul_{\mathbb{F}_{p^k}})\log_2(n)$

The Weil pairing is composed of one Miller Lite, one Full Miller, one inversion in $\mathbb{F}_{p^k}$ and an optional exponentiation to the power $(p^{k/2}-1)$ which allows us to avoid the denominator evaluation in the Miller algorithm. The cost of this exponentiation by $(p^{k/2}-1)$ can be neglected (see section 3.3). The inversion is also due to a Frobenius map (sees section 3.4). Thus, the cost of Weil pairing is given by: $C_W = C_{Lite} + C_{Full} + Mul_{\mathbb{F}_{p^k}}$.

Concerning the Tate pairing, we execute one Lite Miller, and one exponentiation to $(p^k-1)/n$ (more exactly to $\frac{\Phi_k(p)}{n}$, see section 3.3). We have two options: the Lucas sequence method, or a signed sliding windows method for evaluating the exponentiation. In practice, the second approach is more efficient (excepted

for $k=6$). The cost of Lucas sequence method is $C_{Luc}=(Mul_{\mathbb{F}_{p^{k/2}}}+Squ_{\mathbb{F}_{p^{k/2}}})log_2(\frac{\Phi_k(p)}{n})$. While the cost of sliding windows method is $C_{sw}=\left(\frac{log_2(e)}{log_2(p)}+log_2(p)\right)Squ_{G_{\phi_k(p)}}+\left(\frac{log_2(e)}{log_2(p)}\left(2^{r-1}-1\right)+\frac{log_2(e)}{r+2}-1\right)M$, where $e=\frac{\Phi_k(p)}{n}$, and r is the size of the window used, practically, $r=4$.

We recall that for $k=2^i3^j$, we have, using Karatsuba and Toom-Cook, $Mul_{\mathbb{F}_{p^k}}=3^i5^jMul_{\mathbb{F}_p}$, and $Mul_{\mathbb{F}_{p^k}}=3^{i-1}5^jMul_{\mathbb{F}_p}$ (same remark for the squaring).

Case $k=2$: The Lucas sequence method is efficient, the cost is then $C_{Luc}=(Mul_{\mathbb{F}_p}+Squ_{\mathbb{F}_p})log_2(\frac{\Phi_2(p)}{n})$, with $log_2(\frac{\Phi_2(p)}{n})=(\frac{\gamma}{2}-1)b_n$. The cost of the exponentiation is then $C_{Luc}=(\frac{\gamma}{2}-1)(Mul_{\mathbb{F}_p}+Squ_{\mathbb{F}_p})b_n$.

To summarize : the Weil pairing needs $C_W=36Mul_{\mathbb{F}_p}.b_n$, than the Tate pairing needs $C_T=(16+\gamma)Mul_{\mathbb{F}_p}.b_n$. Comparing the two costs, we find that for $\gamma<20$ (i.e. for a security level of 128 bits) Tate is better than Weil, but for higher levels Weil becomes interesting. We can remark that if we use the signed sliding window method, then Weil is more efficient.

Case $k>4$: For $k=6$ and $p\equiv 2 \pmod 9$, we will use the cyclotomic subgroup to exploit the property of the squaring (see last remark of section 3.5). Then for $k=12$ and 24, we will consider the general case of section 3.5. By this way, we can use the improvement of the square in $\mathbb{F}_{p^k}$ from A. Lenstra and M. Stam, and the improvement of the twisted curve of N. Koblitz and A. Menezes. The multiplications are made using Karatsuba and Tom cook methods. Here, the signed sliding windows is the most efficient way to do the exponentiation.

The expressions of the complexities are:

$$C_W = \left((2k+4)Mul_{\mathbb{F}_p}+4Squ_{\mathbb{F}_p}+6Mul_{\mathbb{F}_{p^{k/2}}}+4Squ_{\mathbb{F}_{p^{k/2}}}+2Mul_{\mathbb{F}_{p^k}}+2Squ_{G_{\phi_k(p)}}\right)log_2(n)+Mul_{\mathbb{F}_{p^k}}(4)$$

$$C_T = \left((k+4)Mul_{\mathbb{F}_p}+4Squ_{\mathbb{F}_p}+Mul_{\mathbb{F}_{p^k}}+Squ_{G_{\phi_k(p)}}\right)log_2(n)+C_{sw} \quad (5)$$

where, $C_{sw}=\left(\frac{k}{\gamma}\left(\frac{\gamma}{3}-1\right)+\frac{\gamma}{k}b_n\right)Squ_{G_{\phi_k(p)}}+\left(\frac{\gamma/3-1}{r+2}b_n+\frac{k}{\gamma}(2^{r-1}-1)-1\right)Mul_{\mathbb{F}_{p^k}}$ (see the second part of section 3.3) .

Table 2 presents a comparison between Weil and Tate for the different security levels. Each time, as possible, we had used the different tricks presented (squaring, exponentiation, cyclotomic group, friendly fields), for $k=6,12$ we use Jacobean coordinates but for $k=24$ affine coordinates give a better complexity.

For the given levels of table 2, the Tate pairing is always more efficient than the Weil pairing.

We remarked that the calculus are more efficient for the two pairings in Jacobian coordinates instead of affine (except for $k=24$), which differs from the conclusion of Granger et al.[24] We can see that the difference of the number of operations, between these two pairings, decreases while the security level increases. But presently, security levels higher than 128 or 192 bits are not considered. But that can change... Furthermore, Lite Miller is well known and a lot of works have been done on it, it is not the case of the Full Miller where some tricks can be found. Thus, the challenge between these two pairings is not finished.

4. DISCUSSION ON THE ARITHMETIC POINT OF VIEW: SOME PERSPECTIVES

In their paper,[24] R Granger, D. Page and N. Smart propose to use an interleave Montgomery algorithm in $2t^2+1$ words operations where $t=\log_2 p/w$. It is probably more judicious to evaluate the product (t^2) and the reduction (t^2+t) separately taking into account that for large p Karatsuba or Toom-Cook methods could be appropriate for the multiplications, and, as p should be chosen as a Solinas number, the reduction can be linear reduced to some additions. Furthermore, we don't need to apply a reduction after each product.

k	$Mul_{\mathbb{F}_{p^{k/2}}}$	$Mul_{\mathbb{F}_{p^k}}$	$Squ_{G_{\phi_k(p)}}$
6	$5Mul_{\mathbb{F}_p}$	$15Mul_{\mathbb{F}_p}$	$9Mul_{\mathbb{F}_p}$
12	$15Mul_{\mathbb{F}_p}$	$45Mul_{\mathbb{F}_p}$	$30Mul_{\mathbb{F}_p}$
24	$45Mul_{\mathbb{F}_p}$	$135Mul_{\mathbb{F}_p}$	$108Mul_{\mathbb{F}_p}$

k	security level	Nb op. for Weil	Nb op. for Tate
6	80	18 880	8 180
-	128	30 208	16 428
-	192	45 312	31 592
-	256	60 416	52 456
-	384	90 624	90 184
12	80	49 920	19 600
-	128	79 872	39 320
-	192	119 808	73 480
-	256	159 744	118 640
-	384	239 616	192 000
24	80	158 720	56 000
-	128	253 952	109 400
-	192	380 928	202 600
-	256	507 904	320 800
-	384	761 856	501 200

Table 2. Comparison in number of multiplications in $\mathbb{F}_p$

In fact, the reduction can be made after some additions which don't increase so much the size. That means that, if two products have to be added then the reduction can be made after this addition, like that we win a reduction, and then, we reduce the complexity of the evaluation.

Now, if we count the number of word operations (32 bis), we must take care of the cost of the additions in the fields. In fact, on a processor, an addition takes one cycle, so it is significant in the total cost.

In the same paper,[24] it is claimed that the best multiplications and squarings are obtained with combination of Karatsuba and Toom-Cook. They suppose that the cost in $\mathbb{F}_{p^k}$, for $k = 2^i 3^j$, is equivalent to $3^i 5^j$ times one multiplication in F_p, which is three times the one in $\mathbb{F}_{p^{k/2}}$. This approach could be extend to other values of k with more complex combinations. This last remark could be interesting, they said that a larger k can allow to reduce the bandwidth. In this case, p is smaller and, it could be interesting to consider some works on medium primes.[3] In the other hand, some contributions[5] deal with curves of embedded degree equals to one.

Two other points, they claim that the cost of one squaring is the same than the one of one multiplication in F_p, but generally it is admit that one squaring represents 0.8 multiplication. The second point concerns the inversion, they suppose that one inversion is equivalent to 10 multiplications. This is optimitic when you know that one inversion is between 9 to 40 times costly than one multiplication.

It seems that, as we can chose the curve and the field, it is possible to select a value for n with few bits to one. In this case sliding windows in the Miller Algorithm is not interesting.

Most of the literature deals with finite fields like F_p or F_{2^k}, but some works consider characteristic three fields.[1,9,23] It could be interesting to compare pairings for these different characteristics.

REFERENCES

1. Ahmadi O., Hankerson D., Menezes A.: Software Implementation of Arithmetic in $\mathbb{F}_{3^m}$, WAIFI Conference, Madrid Spain 2007.
2. Barreto, P.: The Pairing-Based Crypto Lounge
   http://paginas.terra.com.br/informatica/paulobarreto/pblounge.html
3. Bajard J.C., Imbert L., Negre Ch.: Arithmetic Operations in Finite Fields of Medium Prime Characteristic Using the Lagrange Representation, IEEE Transactions on Computers, September 2006 (Vol. 55, No. 9) p p. 1167-1177

4. Barreto P., Kim H., Lynn B., Scott M.: Efficient algorithms for pairing-based cryptosystems, Advances in Cryptology CRYPTO 2002, Lecture Notes in Computer Science, 2442 (2002), 354-368.
5. Barreto P., Naehrig M.: Pairing-friendly elliptic curves of prime order, Selected Areas in Cryptography (SAC 2005), LNCS 3897 (2006), 319-331.
6. Barreto P., Lynn B., Scott M.: Efficient implementation of pairing-based cryptosystems, Journal of Cryptology, 17 (2004), 321-334
7. Barreto P., Lynn B., Scott M.: On the Selection of Pairing-Friendly Groups Selected Aeras in Cryptography SAC 2003, LNCS 30006, 2004,17-25
8. Barreto P., Scott M.: Compressed pairings, Advances in Cryptology — Crypto 2004, LNCS 3152, 140-156, http://eprint.iacr.org/2004/032.
9. Beuchat J.L., Brisebarre N., Okamoto E., Shirase M., Takagi T.: A Coprocessor for the Final Exponentiation of the η_T Pairing in Characteristic Three, Proceedings of Waifi 2007, numéro 4547 of LNCD, pages 25-39, 2007.
10. Boneh D., Franklin M.: Identity-based encryption from the Weil pairing. SIAM Journal of Computing, 32, 586?615, 2003
11. Brier E., Joye M.: Point multiplication on elliptic curves through isogenies, AAECC 2003, LNCS., vol. 2643, 2003, 43?50.
12. Blake F., Seroussi G., Smart N. (editors): Advances in Elliptic Curve Cryptography, Series: London Mathematical Society Lecture Note Series (No. 317),Cambridge University Press,2005
13. Ciet, M., Neve, M., Peeters, E., Quisquater, J.J.: Parallel FPGA implementation of RSA with residue number systems– can side-channel threats be avoided? 46th IEEE International Midwest Symposium on Circuits and Systems (2003)
14. Cohen, H., Frey, G. (editors): Handbook of elliptic and hyperelliptic curve cryptography. Discrete Math. Appl., Chapman & Hall/CRC (2006)
15. Diffie W., Hellman M.: directions in cryptography, IEEE Transactions on Information Theory, 22 (1976), 644-654.
16. Duquesne S., Frey G. Background on Pairings, Chapter 6 of Cohen, H., Frey, G.: Handbook of elliptic and hyperelliptic curve cryptography. Discrete Math. Appl., Chapman & Hall/CRC (2006)
17. Duquesne S., Frey G. Implementation of Pairings, Chapter 16 of Cohen, H., Frey, G.: Handbook of elliptic and hyperelliptic curve cryptography. Discrete Math. Appl., Chapman & Hall/CRC (2006)
18. Doche C., Lange T.: Arithmetic of Elliptic Curves Chapter 13 of Cohen, H., Frey, G.: Handbook of elliptic and hyperelliptic curve cryptography. Discrete Math. Appl., Chapman & Hall/CRC (2006)
19. Frey G., Müller M., Rück H.G.: The Tate Pairing and the Discrete Logarithm Applied to Elliptic Curve Cryptosystems, IEEE Transactions Inf. Theory, 45, 1717-1719;1999
20. Fleischmann P., Paar C., Soria-Rodriguez P.: Fast Arithmetic for Public-Key Algorithms in Galois Fields with Composite Exponents, IEEE Transactions on Computers, vol. 48, no. 10, pp. 1025-1034, October, 1999.
21. Frey G., Rück H.G.: A Remark Concerning m-divisibility Constructions and the Discrete Logarithmic problem in the Divisor Class Group of Curves, in Math. comp., 62, 865-874, 1994.
22. Galbraith S.: K.G.: Pairings, Chapter IX, Advances in Elliptic Curve Cryptography, F. Blake and G. Seroussi and N. Smart editors, Series: London Mathematical Society Lecture Note Series (No. 317),Cambridge University Press,2005
23. Granger R., Page D., Stam M.: Hardware and Software Normal Basis Arithmetic for Pairing-Based Cryptography in Characteristic Three. IEEE Transactions on Computers,volume 54(7): 852–860, July 2005
24. Granger R., Page D., Smart N.: High security pairing-based cryptography revisited. Proceedings ANTS-7, Springer LNCS 4096, pp 480-494, (2006)
25. Joux A.: one round protocol for tripartite Diffie-Hellman, Algorithmic Number Theory: Fourth International Symposium, Lecture Notes in Computer Science, 1838 (2000), 385-393. Full version: Journal of Cryptology, 17 (2004), 263-276

26. Koblitz N.: Elliptic curve cryptosystems, Mathematics of Computation, Vol. 48, 1987, 203-209.
27. Koblitz N., Menezes A. J.: Pairing-based cryptography at high security levels, Proceedings of the Tenth IMA International Conference on Cryptography and Coding, Springer-Verlag, LNCS 3796, 2005, 13-36;
28. Lange T.: Mathematical Background of Pairings, http://www.hyperelliptic.org/tanja/vortraege/pairings_lange.ps, ECRYPT PhD Summer School on Emerging Topics in Cryptographic Design and Cryptanalysis, Samos, Greece, 2007.
29. Lenstra A., Stam M.: Efficient Subgroup Exponentiation in Quadratic and Sixth Degree Extensions, Cryptographic Hardware and Embedded Systems, CHES 2002, LNCS 2523 pp. 318-332.
30. Menezes A.: An introduction to pairing-based cryptography Notes from lectures given in Santander, Spain, 2005 http://www.cacr.math.uwaterloo.ca/ ajmeneze/publications/pairings.pdf
31. Menezes A.: Supersingular Elliptic Curves in Cryptography, Invited talk, Pairing 2007, Tokyo Japan.
32. Miller V.: Short Programs for Functions on Curves, IBM, Thomas J. Watson Research Center, 1986 http://crypto.stanford.edu/miller/miller.pdf
33. Miller V.: Use of Elliptic Curves in Cryptography Advances in Cryptology-Crypto 85 pages 417-426 Vol. 218 of LNCS 1986.
34. Miller V.: The Weil pairing and its efficient calculation, J. Cryptology, 17 (2004), 235-261.
35. Montgomery, P.L.: Speeding the Pollard and elliptic curve methods of factorization. Math. Comp. 48:177 (1987) 243–164
36. MenezesA., Okamoto T. and Vanstone S.A.: Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field, IEEE Trans. Inf . Theory 39, numéro 5, pages 1639-1646, 1993.
37. Mumford D.: Abelian Varietes, Oxford University Press, 1974
38. Pairing 2005 in Dublin Ireland, http://pic.computing.dcu.ie/ Pairing 2007 in Tokyo Japan, http://www.pairing-conference.org/
39. Paterson K.G.: Cryptography from Pairings, Chapter X, Advances in Elliptic Curve Cryptography, F. Blake and G. Seroussi and N. Smart editors, Series: London Mathematical Society Lecture Note Series (No. 317),Cambridge University Press,2005
40. Shamir A.: Identity BasedCryptosystems and Signature Schemes, Advances in Cryptology Crypto '84, LNCS, Vol. 196, pp 47-53, 1984
41. Scott M.: Computing the Tate pairing, Topics in Cryptology?CT-RSA 2005, LNCS 3376 (2005), 293-304.
42. Thomé E.: Applications algorithmiques des courbes elliptiques (in French), http://www.loria.fr/ thome/php/dl.php/publis/slides/ejc_20070319.pdf, École jeune chercheurs en informatique mathématique (EJC IM), Nancy, mars 2007.