ANR Project ECLIPSES

Elliptic Curve Leakage-Immune Processing for Secure Embedded Systems

D2.1

Promising Algorithms for Pairing Computations

Contributor(s)

Matthieu Rivain – CryptoExperts

Due date of deliverable:	T0+6
Actual submission date:	November 4, 2010
ECLIPSES partner in Charge:	CryptoExperts

Confidentiality level: **Restricted to ECLIPSES** **Release:** 1.0

History

Version	Date	Author	Modification
1.00	09/09/2010	Matthieu Rivain	Initial version
1.01	29/10/2010	Matthieu Rivain	Minor corrections and addition of Section 6

ECLIPSES Partners

Start date of project: 2010, January 21 Duration: 3 years

Contents

1 Introduction **1**

2 Mathematical background on elliptic curves **2**
2.1 Basic definition 2
2.2 Addition law 3
2.3 Group structure and r-torsion 6
2.4 Supersingular vs. ordinary curves 7
2.5 Twist of an elliptic curve 8
2.6 Frobenius endomorphism and trace map 10
2.7 Function field and divisors 12

3 Pairings **15**
3.1 Basic definition 15
3.2 The Weil Pairing 16
3.3 The Tate Pairing 17
3.4 Pairings over cyclic subgroups 19
3.5 Pairing-friendly elliptic curves 20

4 Promising pairing-based protocols **21**
4.1 Hardness assumptions 21
4.2 Boneh-Franklin identity-based encryption scheme 23
4.3 Boneh-Lynn-Shacham short signature scheme 24
4.4 Joux one-round tripartite key agreement 26

5 Pairing computation algorithms **28**
5.1 Miller's algorithm 28
5.2 Extension field arithmetic 31
5.3 Tate pairing optimizations 34
5.4 The Eta and Ate pairings 38
5.5 Generalizations and optimal pairings 41

6 Summary and recommendations **43**

1 Introduction

The purpose of this document is to give an overview of promising algorithms for pairing computation in cryptography. In a first place, we introduce some mathematical background on elliptic curves which is necessary to the understanding of pairings used in cryptography. Then we formally define what is a pairing and we recall the different types of pairing consider in cryptography. We describe the Weil and the Tate pairings, and we focus on restrictions of these pairings such as used in practice. Afterwards we address pairing-based cryptography: we recall usual hardness assumptions and we describe three different pairing-based cryptographic protocols. Finally, we review efficient algorithms for the computation of pairings, including Miller's algorithm, the extension field arithmetic, and several optimizations and variants of the Tate pairing.

2 Mathematical background on elliptic curves

In this section, we give necessary mathematical background on elliptic curves for understanding pairing-based protocols and pairing computation algorithms. We will restrict our description to elliptic curves over finite fields which is sufficient in the context of cryptography. We use basic notions of algebra such as *(cyclic) group*, *finite field*, *field characteristic*, *field extension*, etc. Background about these notions can be found in [LN97]. The reader is further referred to [BIPV10] for details about elliptic curve arithmetic and elliptic curve cryptography.

2.1 Basic definition

Definition 2.1 (Elliptic Curve)**.** An *elliptic curve* E over a finite field $\mathbb{F}_q$ of characteristic $p > 3$ is defined by the following *short Weierstrass equation*:

$$E : y^2 = x^3 + ax + b \tag{1}$$

where $a, b \in \mathbb{F}_q$ and $\Delta = -16(4a^3 + 27b^2) \neq 0$ (Δ is called *discriminant* of E). For every extension K of $\mathbb{F}_q$, the *K-rational points of E* is the set $E(K)$ of points $(x, y) \in K \times K$ satisfying (1) together with the *point at infinity* denoted $\mathcal{O}$. The points in $E(\mathbb{F}_q)$ are simply called the *rational points of E*.

In the following, we will denote by $p_E \in \mathbb{F}_q[x, y]$ the polynomial in the equation of E:

$$p_E(x, y) = x^3 + ax + b - y^2 \ ,$$

which is such that $P \in E(K)\backslash\{\mathcal{O}\}$ if and only if $p_E(P) = 0$.

Remark. This definition can be extended to finite fields of characteristic $p \in \{2, 3\}$; only (1) and Δ change depending on whether $p = 2$ or $p = 3$ and on whether the elliptic curve is *ordinary* or *supersingular* (cf. Section 2.4). For completeness, we summarize the different possibilities in Table 1.

Table 1: Equations and discriminants of elliptic curve over $\mathbb{F}_{p^m}$.

p	ordinary curves	supersingular curves
2	$E : y^2 + xy = x^3 + ax^2 + b$ $\Delta = b$	$E : y^2 + cy = x^3 + ax + b$ $\Delta = c^4$
3	$E : y^2 + xy = x^3 + ax^2 + b$ $\Delta = -a^3 b$	$E : y^2 + cy = x^3 + ax + b$ $\Delta = -a^3$
> 3	$E : y^2 = x^3 + ax + b$ $\Delta = -16(4a^3 + 27b^2)$ (Definition 2.1)	

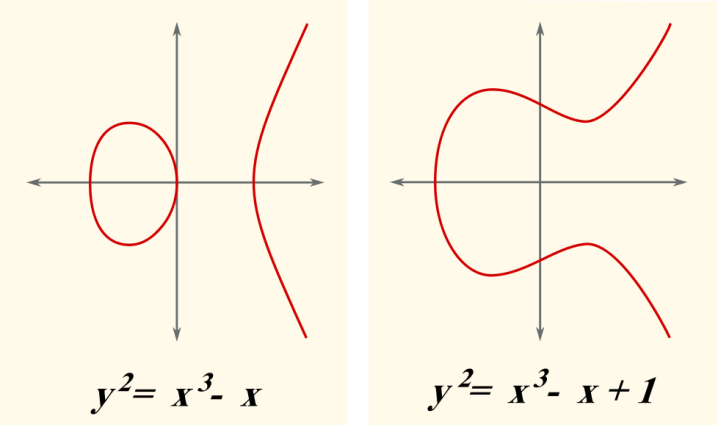

Figure 1: Elliptic curves over $\mathbb{R}$ [Wik].

Although we only focus on elliptic curves over finite fields they can be defined over any field. As an illustration, Fig. 1 represents the graphs of two different elliptic curves over $\mathbb{R}$.

It is clear from Definition 2.1 that for every pair of natural integers $m_1 < m_2$ we have $E(\mathbb{F}_{q^{m_1}}) \subset E(\mathbb{F}_{q^{m_2}})$ (as we have $\mathbb{F}_{q^{m_1}} \subset \mathbb{F}_{q^{m_2}}$). Let us recall that the *algebraic closure* of a finite field $\mathbb{F}_q$, denoted $\overline{\mathbb{F}_q}$, is the field containing all the extensions of $\mathbb{F}_q$, that is $\overline{\mathbb{F}_q} = \bigcup_{n=1}^{\infty} \mathbb{F}_{q^n}$. Then we naturally have

$$E\left(\overline{\mathbb{F}_q}\right) = \bigcup_{n=1}^{\infty} E(\mathbb{F}_{q^n}) \, .$$

The next theorem gives an interval for the number of rational points of an elliptic curve.

Theorem 2.1 (Hasse)**.** *Let E be an elliptic curve defined over $\mathbb{F}_q$. Then we have $\#E(\mathbb{F}_q) = q + 1 - t$ where t is called the* trace *of E and satisfies:*

$$-2\sqrt{q} \leq t \leq 2\sqrt{q} \, . \tag{2}$$

Since E is defined over $\mathbb{F}_q$, E is also defined over any extension $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$, and we can deduce from the above theorem that for every m we have:

$$q^m + 1 - 2q^{m/2} \leq \#E(\mathbb{F}_{q^m}) \leq q^m + 1 + 2q^{m/2} \, .$$

2.2 Addition law

Let E be an elliptic curve defined over $\mathbb{F}_q$. For every extension K of $\mathbb{F}_q$, the set of K-rational points of E can be provided with an addition law:

$$\begin{aligned} E(K) \times E(K) &\longrightarrow E(K) \\ (P, Q) &\longmapsto P + Q \end{aligned}$$

which is associative and commutative, has identity element $\mathcal{O}$ and, for which every $P \in E(K)$, has an inverse element $-P \in E(K)$. This implies that $(E(K), +, \mathcal{O})$ has a finite Abelian group structure.

The addition law is given by the so-called *chord-and-tangent rule*. Geometrically, this rule says that drawing a line which intersects the curve at several points implies that the sum of these points equals the identity $\mathcal{O}$. The points are counted with *multiplicity* meaning that if the line is a tangent of E at P then it intersects the curve twice at P. Four different cases occur:

1. The line intersects the curve at three distinct points P, Q and R. We have $P + Q + R = \mathcal{O}$.
2. The line intersects the curve at P and is tangent to the curve at Q. We have $P + Q + Q = \mathcal{O}$.
3. The line intersects the curve at two distinct points P and Q. We have $P + Q = \mathcal{O}$.
4. The line is tangent to the curve at a point P and does not further intersect the curve. We have $P + P = \mathcal{O}$.

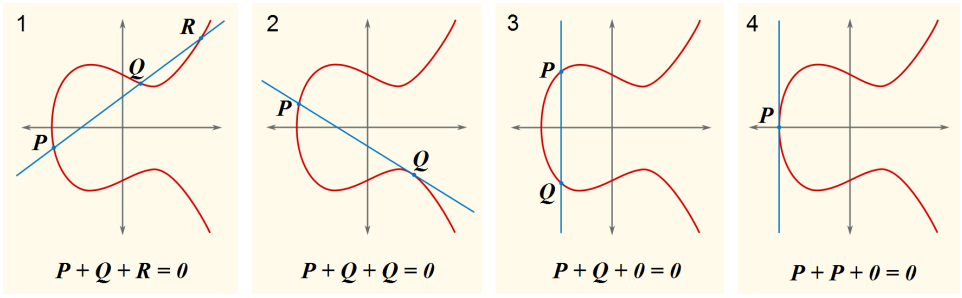

Figure 2: The chord-and-tangent rule [Wik].

These different cases are illustrated in Fig. 2. According to this rule, we deduce that the inverse of a point $P = (x_1, y_1)$ is simply its reflection by the x-axis *i.e.* $-P = (x_1, -y_1)$ (see case 3). Stating that any line intersects the curve in three points, we obtain a geometric interpretation of the point at infinity: it is the point where all the vertical lines intersect (*i.e.* at infinity). We then have that every vertical line intersects the curve in P, $-P$ and $\mathcal{O}$ which is consistent with $P - P + \mathcal{O} = \mathcal{O}$ (see case 3 and 4). Note that case 4 corresponds to $P = -P$ or equivalently $P + P = \mathcal{O}$, that is P is a point of order 2 of $E(K)$.

The chord-and-tangent rule also enables to efficiently compute the sum of two points P and Q. Observe for instance in case 1 that $P + Q + R = \mathcal{O}$ implies $P + Q = -R$. Therefore, adding two points P and Q (when $P \neq \mathcal{O}$, $Q \neq \mathcal{O}$ and $P \neq -Q$) consists in drawing a line through them (or the tangent of E at P if $P = Q$) and taking $P + Q$ as the reflection by the x-axis of the third point intersected by the line. From this principle, we can derive addition and doubling formulae as detailed hereafter.

Adding two points. Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ with $P \neq Q$ and $P, Q \neq$

$\mathcal{O}$. Let

$$L : y = \lambda x + \beta \tag{3}$$

be the line through P and Q. We have:

$$\lambda = \frac{y_1 - y_2}{x_1 - x_2} \text{ and } \beta = y_1 - \lambda x_1 \, . \tag{4}$$

The intersection of L and E is the set of points $(x, y) \in K \times K$ satisfying both (3) and (1). In particular, the x-coordinates of those points satisfies $(\lambda x + \beta)^2 = x^3 + ax + b$ which gives:

$$p(x) := x^3 - \lambda^2 x^2 - 2\beta\lambda x + ax + b - \beta^2 = 0 \, . \tag{5}$$

Since $P, Q \in L \cap E$, x_1 and x_2 are both solution of the previous equation (*i.e.* they are roots of p) hence we deduce:

$$p(x) = (x - x_1)(x - x_2)(x - x_3) \tag{6}$$

where x_3 is the third root of p. Defining $y_3 = -\lambda x_3 - \beta$, we get $E \cap L = \{P, Q, (x_3, -y_3)\}$ and $P + Q = (x_3, y_3)$. Finally, we can efficiently compute $P + Q$, since from (5) and (6) we have $x_1 + x_2 + x_3 = \lambda^2$ which gives:

$$x_3 = \lambda^2 - x_1 - x_2 = \left(\frac{y_1 - y_2}{x_1 - x_2}\right)^2 - x_1 - x_2 \, , \tag{7}$$

and:

$$y_3 = -\lambda x_3 - \beta = \left(\frac{y_1 - y_2}{x_1 - x_2}\right)(x_3 - x_1) - y_1 \, . \tag{8}$$

Doubling a point. When $P = Q \neq \mathcal{O}$ the sum $P + Q = [2]P$ is computed similarly than when $P \neq Q$ except that we take the tangent of E at P instead of the line through P and Q. The slope of this tangent satisfies:

$$\lambda = -\frac{\frac{\partial p_E}{\partial x}(x_1, y_1)}{\frac{\partial p_E}{\partial y}(x_1, y_1)} = \frac{3x_1^2 + a}{2y_1} \, , \tag{9}$$

which gives:

$$x_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1 \, , \tag{10}$$

and:

$$y_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)(x_3 - x_1) - y_1 \, . \tag{11}$$

It is clear that the chord-and-tangent addition law is commutative *i.e.* $P + Q = Q + P$ (whatever the order of P and Q around the '+', the third points intersected by the line through P and Q is always the same). The associativity of the law is more complicated to show. The interested reader is referred to [Sil86, Joy95] for a demonstration.

2.3 Group structure and r-torsion

In the following, we will denote by $[k]P$ the *scalar multiplication* of a point P by an integer k which is defined as:

$$[k]P = \underbrace{P + P + \cdots + P}_{k \text{ times}} .$$

Let us recall that the cardinal of a finite Abelian group $(\mathbb{G}, +, \mathcal{O})$ is also called its *order*, denoted $\#\mathbb{G}$, and that the *order* of an element $P \in \mathbb{G}$ is the minimum positive integer n such that $[n]P = \mathcal{O}$. Let us further recall that, every finite Abelian group $\mathbb{G}$ is isomorphic to $\mathbb{Z}_{k_1} \oplus \mathbb{Z}_{k_2} \oplus \cdots \oplus \mathbb{Z}_{k_m}$ for some k_i's satisfying $\prod_i k_i = \#\mathbb{G}$ and $1 < k_1 \mid k_2 \mid \cdots \mid k_m$ (where $\mathbb{Z}_k$ denotes the additive group $(\mathbb{Z}/k\mathbb{Z}, +, 0)$ of integers modulo k and '$\oplus$' denotes the direct sum operator). This implies that every element in $\mathbb{G}$ can be expressed as a linear combination of m distinct elements P_1, P_2, …, P_m of $\mathbb{G}$ having order k_1, k_2, …, k_m respectively. As stated in the next proposition, the structure of $(E(\mathbb{F}_q), +, \mathcal{O})$ can be further specified.

Proposition 2.2. *Let E be an elliptic curve defined over $\mathbb{F}_q$, then:*

$$E(\mathbb{F}_q) \simeq \mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_2}$$

where $n_1 \mid n_2$ and $n_1 \mid (q-1)$.

If E is defined over $\mathbb{F}_q$, it is also defined over any extension $\mathbb{F}_{q^m}$ and we can deduce that for every $m \in \mathbb{N}$ we have $E(\mathbb{F}_{q^m}) \simeq \mathbb{Z}_{n'_1} \oplus \mathbb{Z}_{n'_2}$ where $n'_1 \mid n'_2$ and $n'_1 \mid (q^m - 1)$.

For cryptographic applications and in particular for pairing-based cryptography, we shall make use of a subgroup of $E\big(\overline{\mathbb{F}_q}\big)$ which, for some integer r, is known as the *r-torsion* of E.

Definition 2.2 (r-torsion)**.** Let E be an elliptic curve defined over a field $\mathbb{F}_q$. A point P of E is said to be of *r-torsion* if it satisfies $[r]P = \mathcal{O}$ (*i.e.* the order of P divides r). Let K be an extension of $\mathbb{F}_q$, the group of r-torsion points in $E(K)$ is denoted $E(K)[r]$. The group $E\big(\overline{\mathbb{F}_q}\big)[r]$ is simply denoted $E[r]$ and is called the *r-torsion of E*.

The next theorem gives the group structure of the r-torsion.

Theorem 2.3 (Cor. III.6.4 [Sil86])**.** *Let E be an elliptic curve defined over $\mathbb{F}_q$ and let $r \in \mathbb{N}^*$. If $\gcd(r, q) = 1$, then $E[r] \simeq \mathbb{Z}_r \oplus \mathbb{Z}_r$. Otherwise we have either $E[p^e] \simeq \mathbb{Z}_{p^e}$ for every $e \geq 1$, or $E[p^e] \simeq \{\mathcal{O}\}$ for every $e \geq 1$.*

A natural question to ask while working with the r-torsion of an elliptic curve is: which extension K of $\mathbb{F}_q$ must be considered to get the whole r-torsion of $E[r]$ included in $E(K)$? This motivates the notion of *embedding degree*.

Let E be an elliptic curve defined over $\mathbb{F}_q$ and let r be an integer such that E has $\mathbb{F}_q$-rational points of order r and $\gcd(r, q) = 1$. The *embedding degree* of E

with respect to r is the extension degree $[\mathbb{F}_q(\mu_r) : \mathbb{F}_q]$ where μ_r is the set of rth roots of unity in $\overline{\mathbb{F}_q}$. Then we have the equivalence:

$$\mathbb{F}_q(\mu_r) = \mathbb{F}_{q^k} \iff r \mid q^k - 1$$

from which we deduce the following equivalent definition.

Definition 2.3 (Embedding degree)**.** Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$ and let r be an integer such that $r \mid \#E(\mathbb{F}_q)$ and $\gcd(r, q) = 1$. The *embedding degree* of $E(\mathbb{F}_q)$ with respect to r is the least positive integer k such that $r \mid q^k - 1$.

Proposition 2.4 ([BK98])**.** *Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$ and let r be a prime such that $r \mid \#E(\mathbb{F}_q)$, $\gcd(r, q) = 1$, and $r \nmid q - 1$. Then $E[r] \subset E(\mathbb{F}_{q^k})$ if and only if $r \mid q^k - 1$.*

Proposition 2.4 implies that if the embedding degree k of E with respect to some prime r is greater than 1, then $\mathbb{F}_{q^k}$ is the smallest extension of $\mathbb{F}_q$ such that $E(\mathbb{F}_{q^k})$ contains the entire r-torsion of E.

To summarize and assuming r to be prime and co-prime to q, we have

$$E[r] = E(\mathbb{F}_{q^k})[r] \simeq \mathbb{Z}_r \oplus \mathbb{Z}_r \ .$$

Namely, (i) the points of $E[r]$ have coordinates belonging to $\mathbb{F}_{q^k}$, and (ii) the r-torsion is a 2-dimensional vector space over $\mathbb{F}_r$. Every point $P \in E[r]$ generate a proper subgroup/subspace $\langle P \rangle$ of $E[r]$ which is a cyclic group of order r. As every subgroup contains r points and as all the subgroups have one single point in common (which is $\mathcal{O}$), there exist $r + 1$ subgroups. An example is the subgroup of rational r-torsion points $E(\mathbb{F}_q)[r]$. Eventually, any $(P, Q) \in E[r] \times E[r]$ with $P \notin \langle Q \rangle$ (or equivalently $Q \notin \langle P \rangle$) is a basis of $E[r]$ and every point $R \in E[r]$ can be expressed as $R = aP + bQ$ where $a, b \in \mathbb{Z}_r$.

2.4 Supersingular vs. ordinary curves

Definition 2.4. Let E be an elliptic curve defined over $\mathbb{F}_q$ of characteristic p. E is said to be *supersingular* if it satisfies $\#E(\mathbb{F}_q) \equiv 1 \bmod p$ (*i.e.* the trace of E is a multiple of p). Otherwise E is said to be *ordinary*.

In other words, an elliptic curve E defined over $\mathbb{F}_q$ is supersingular if and only if its trace $t = q + 1 - \#E(\mathbb{F}_q)$ is a multiple of the characteristic of $\mathbb{F}_q$.

One of the main feature of supersingular elliptic curves which makes them interesting for pairing-based cryptography is that they have small embedding degrees (≤ 6). As we will see later, this feature makes them *pairing-friendly*. We summarize in Table 2, all possibilities for the embedding degree k (with respect to any $r|\#E(\mathbb{F}_q)$), the cardinal q of the base field, the trace t, the number of $\mathbb{F}_q$-rational points and the number of $\mathbb{F}_{q^k}$-rational points of a supersingular elliptic curve defined over $\mathbb{F}_q$ (p denotes any prime number and n any positive integer).

Table 2: Possible parameters of supersingular elliptic curves.

k	q	t	$\#E(\mathbb{F}_q)$	$\#E(\mathbb{F}_{q^k})$
1	p^{2n}	$\pm 2\sqrt{q}$	$q \mp 2\sqrt{q} + 1$	$(q \mp 1)^2$
2	p^{2n+1}	0	$q + 1$	$(q + 1)^2$
2	p^{2n} ($p \not\equiv 1[4]$)	0	$q + 1$	$(q + 1)^2$
3	p^{2n} ($p \not\equiv 1[3]$)	$\pm\sqrt{q}$	$q \mp \sqrt{q} + 1$	$(q^{3/2} \mp 1)^2$
4	2^{2n+1}	$\pm\sqrt{2q}$	$q \mp \sqrt{2q} + 1$	$(q^2 + 1)^2$
6	3^{2n+1}	$\pm\sqrt{3q}$	$q \mp \sqrt{2q} + 1$	$(q^3 + 1)^2$

It can be noticed from Table 2 that $\#E(\mathbb{F}_{q^k})$ is always a square. We actually have $E(\mathbb{F}_{q^k}) \simeq \mathbb{Z}_{\sqrt{N}} \times \mathbb{Z}_{\sqrt{N}}$ where $N = \#E(\mathbb{F}_{q^k})$.

Finally, we introduce the notion of *distortion map* which is useful in cryptography.

Definition 2.5 (distortion map)**.** Let E be an elliptic curve defined over $\mathbb{F}_q$. A *distortion map* of E is an endomorphism $E\big(\overline{\mathbb{F}_q}\big) \to E\big(\overline{\mathbb{F}_q}\big)$ which maps a rational point (*i.e.* a point in $E(\mathbb{F}_q)$) to a non-rational point (*i.e.* a point in $E\big(\overline{\mathbb{F}_q}\big)\backslash E(\mathbb{F}_q)$).

Theorem 2.5. *Let E be an elliptic curve defined over $\mathbb{F}_q$. Then E has a distortion map if and only if E is supersingular.*

The necessity has been proven in [Ver01, Ver04]. In [GR04], the authors show how to construct an efficiently computable distortion map mapping a point of $E(\mathbb{F}_q)[r]$ to a non-rational point of $E(\mathbb{F}_{q^k})[r]$ (where k is the embedding degree of E w.r.t. r). In particular, this result implies the sufficiency of the previous theorem statement.

As we will see later, an *efficiently computable* isomorphism

$$\phi : E(\mathbb{F}_q)[r] \longrightarrow \mathbb{G} \subset E(\mathbb{F}_{q^k})[r]\backslash E(\mathbb{F}_q)[r]$$

is useful to construct a *symmetric pairing* over $E(\mathbb{F}_q)[r]$. If E is a supersingular elliptic curve, then such an isomorphism can be taken as the restriction of a distortion map to $E(\mathbb{F}_q)[r]$. If E is ordinary, no such efficient isomorphism is known and it is currently believed that no such efficient isomorphism exists.

2.5 Twist of an elliptic curve

Let $E : y^2 = x^3 + ax + b$ be an elliptic curve defined over $\mathbb{F}_q$ (of characteristic greater than 3). The *j-invariant of E* is defined as:

$$j(E) = 1728\frac{4a^3}{4a^3 + 27b^2} .$$

The j-invariant is a key notion in elliptic curve arithmetic as it can be shown that an elliptic curve E' is isomorphic to E, namely there exits an isomorphism $\phi : E'\big(\overline{\mathbb{F}_q}\big) \to E\big(\overline{\mathbb{F}_q}\big)$, if and only if $j(E') = j(E)$. By definition of the j-invariant,

one can indeed check that if $j(E') = j(E)$, then there exits $\alpha \in \mathbb{F}_q$ such that E' satisfies:

$$E' : y^2 = x^3 + \frac{a}{\alpha^2}x + \frac{b}{\alpha^3} \,. \tag{12}$$

Then we have $(x, y) \in E'\big(\overline{\mathbb{F}_q}\big)$ if and only if $(\alpha x, \alpha^{3/2} y) \in E\big(\overline{\mathbb{F}_q}\big)$. In other words, E' is isomorphic to E by:

$$\phi : E'\big(\overline{\mathbb{F}_q}\big) \to E\big(\overline{\mathbb{F}_q}\big) : (x, y) \mapsto (\alpha x, \alpha^{3/2} y) \,. \tag{13}$$

In the sequel, we shall consider the following equivalence relation: $E \sim E'$ if and only if there exists an isomorphism between E and E', which is defined over $\mathbb{F}_q$. If two elliptic curves are equivalent then we have $E'(\mathbb{F}_{q^m}) \simeq E(\mathbb{F}_{q^m})$ for every m (and in particular $\#E(\mathbb{F}_{q^m}) = \#E'(\mathbb{F}_{q^m})$). Namely E and E' have the exact same group structure over any extension of $\mathbb{F}_q$.

In (13), if α is a square over $\mathbb{F}_q$ then ϕ is defined over $\mathbb{F}_q$ and $E \sim E'$. On the other hand, if α is a non-square over $\mathbb{F}_q$ then $\alpha^{3/2}$ lies in $\mathbb{F}_{q^2}$ but not in $\mathbb{F}_q$. In that case ϕ is not defined over $\mathbb{F}_q$, it is only defined over even-degree extensions of $\mathbb{F}_q$. E' is then called the *twist* of E (of degree 2) and $E'(\mathbb{F}_{q^m}) \simeq E(\mathbb{F}_{q^m})$ if and only if m is even.

Every elliptic curve has a twist of degree 2 whose equation can be obtained from (12) by picking any $\alpha \in \mathbb{F}_q$ which is not a square (such an α always exists as q is odd). It can then be shown that all degree-2 twists of E are isomorphic one of each other over $\mathbb{F}_q$. This implies that there exists one single degree-2 twist of an elliptic curve E modulo $\sim$.

More generally, we have the following definition of a twist of any degree.

Definition 2.6. Let E and E' be two elliptic curves defined over $\mathbb{F}_q$. E' is called a *twist of degree d of E* if there exists an isomorphism $\phi_d : E'\big(\overline{\mathbb{F}_q}\big) \to E\big(\overline{\mathbb{F}_q}\big)$ defined over $\mathbb{F}_{q^d}$ and d is minimal.

Analogously to the degree-2 case, a twist E' of E of degree d is such that $E'(\mathbb{F}_{q^m}) \simeq E(\mathbb{F}_{q^m})$ if and only if $d \mid m$. The following proposition gives the number of twists of an elliptic curve (including itself as twist of degree 1).

Proposition 2.6 (Prop. X.5.4 [Sil86])**.** *Let E be an elliptic curve defined over $\mathbb{F}_q$ (of characteristic greater than 3). The set of twists of E is canonically isomorphic to $\mathbb{F}_q^*/(\mathbb{F}_q^*)^d$ where:*

$$d = \begin{cases} 6 & \text{if } j(E) = 0 \,, \\ 4 & \text{if } j(E) = 1728 \,, \\ 2 & \text{otherwise.} \end{cases}$$

Note that if $\gcd(d, q-1) = e$ then $\mathbb{F}_q^*/(\mathbb{F}_q^*)^d = \mathbb{F}_q^*/(\mathbb{F}_q^*)^e$ contains e distinct classes and the set μ_e of eth roots of unity is included in $\mathbb{F}_q^*$. Further note that $x \mapsto x^{(q-1)/e}$ defines a group isomorphism from $\mathbb{F}_q^*/(\mathbb{F}_q^*)^e$ to μ_e (in particular, $\mathbb{F}_q^*/(\mathbb{F}_q^*)^e$ is isomorphic to $\mathbb{Z}_e$). In the following we shall assume [1] that d divides $q-1$, which implies $\mathbb{F}_q^*/(\mathbb{F}_q^*)^d \simeq \mathbb{Z}_d$ and $\mu_d \subset \mathbb{F}_q^*$.

1. If d does not divide $q-1$, then d can be replaced by $e = \gcd(d, q-1)$ and the results remain the same.

Let $\alpha \in \mathbb{F}_q^*$ and denote $\overline{\alpha} \in \mathbb{F}_q^*/(\mathbb{F}_q^*)^d$ the class of α (*i.e.* for every $\beta \in \overline{\alpha}$, there exists $\lambda \in \mathbb{F}_q^*$ such that $\beta = \alpha\lambda^d$). According to Proposition 1, every class $\overline{\alpha}$ can be associated to one and one single twist of E modulo $\sim$ (the elliptic curve E being itself associated to the identity class $\overline{1}$). To see this, we can proceed as for the degree-2 case detailed above. Let $\xi \in \mathbb{F}_{q^d}$ be a dth root of α. Then the elliptic curve E' given by:

$$E' : y^2 = x^3 + \frac{a}{\xi^4}x + \frac{b}{\xi^6} \ . \tag{14}$$

is a twist of E with the following isomorphism:

$$[\xi] : E'(\overline{\mathbb{F}_q}) \to E(\overline{\mathbb{F}_q}) : (x, y) \mapsto (\xi^2 x, \xi^3 y) \ .$$

Note that for $d = 2$, we retrieve the twist of degree 2 given by (12) and $[\xi]$ equals the isomorphism given in (13). Also note that, accordingly to Proposition 2.6, we can verify that if E' is a twist of degree 6 then $a = 0$ (otherwise E' would not be defined over $\mathbb{F}_q$ as $\xi^4 \notin \mathbb{F}_q$), and that if E' is a twist of degree 4 then $b = 0$ (otherwise E' would not be defined over $\mathbb{F}_q$ as $\xi^6 \notin \mathbb{F}_q$).

Let us now show that a class $\overline{\alpha}$ yields one and only one twist modulo $\sim$ whatever the choice of $\alpha \in \overline{\alpha}$ and of $\xi \in \sqrt[d]{\alpha}$. On the one hand, choosing two different dth roots ξ_1 and ξ_2 of α yield the same twist modulo $\sim$ as $\xi_1^d = \xi_2^d = \alpha$ implies that $\xi_1 = \xi_2\zeta$ where $\zeta \in \mu_d \subset \mathbb{F}_q$. So the isomorphism $(x, y) \mapsto (\zeta^2 x, \zeta^3 y)$ maps the curve obtained with ξ_1 to the one obtained with ξ_2 and it is defined over $\mathbb{F}_q$. On the other hand, any dth root of $\alpha' \in \overline{\alpha}$ yields the same twist modulo $\sim$ than a dth root of α since there exists $\lambda \in \mathbb{F}_q^*$ such that $\alpha' = \alpha\lambda^d$. This implies that for every dth root ξ' of α' there exists a dth root ξ of α such that $\xi' = \lambda\xi$ and $(x, y) \mapsto (\lambda^2 x, \lambda^3 y)$ defines an isomorphism over $\mathbb{F}_q$ between the two curves.

We further have the interesting property that the degree of E' is the order of $\overline{\alpha}$ in $\mathbb{F}_q^*/(\mathbb{F}_q^*)^d$ (or equivalently the order of $\alpha^{(q-1)/d}$ in μ_d). Indeed, $[\xi]$ is defined over $\mathbb{F}_{q^m}$ if and only if $\xi \in \mathbb{F}_{q^m}^*$. Let $\mathrm{ord}(\overline{\alpha}) = m$, then there exists $\lambda \in \mathbb{F}_q^*$ such that $\alpha^m = \lambda^d$ that is $\xi^m = \zeta\lambda \in \mathbb{F}_q$ for some $\zeta \in \mu_d$. It follows that $\xi \in \mathbb{F}_{q^m}^*$.

Finally we have the following useful proposition.

Proposition 2.7 ([HSV06])**.** *Let E be an elliptic curve defined over $\mathbb{F}_q$ (of characteristic greater than 3) which has a twist of degree d. Let r be a prime such that $r \parallel E(\mathbb{F}_q)$ and $r^2 \parallel E(\mathbb{F}_{q^k})$ with k minimal and $d \mid k$. Denote $m = k/d$. Then E has a unique twist E' of degree d such that $r \parallel E'(\mathbb{F}_{q^m})$.*

2.6 Frobenius endomorphism and trace map

Definition 2.7 (Frobenius endomorphism)**.** Let E be an elliptic curve defined over $\mathbb{F}_q$. The *Frobenius endomorphism* over E is defined as:

$$\begin{aligned} \Phi_q \ : \ E\big(\overline{\mathbb{F}_q}\big) &\longrightarrow E\big(\overline{\mathbb{F}_q}\big) \\ (x, y) &\longmapsto (x^q, y^q) \end{aligned}$$

The mapping $\Phi_q^k :\mapsto (x^{q^k}, y^{q^k})$ is further called the *q^k-power Frobenius endomorphism*.

The fact that (x^q, y^q) belongs to $E(\overline{\mathbb{F}_q})$ can be shown as follows. Let $p_E(x, y)$ be the polynomial in the equation of E. We have:

$$(x, y) \in E(\overline{\mathbb{F}_q}) \iff p_E(x, y) = 0 \iff p_E(x, y)^q = 0 \, .$$

Let $c_{ij} \in \mathbb{F}_q$ denote the coefficients of p_E such that $p_E(x, y) = \sum_{ij} c_{ij} x^i y^j$. We have:

$$p_E(x, y)^q = \sum_{ij} c_{ij}^q x^{iq} y^{jq} = \sum_{ij} c_{ij} x^{iq} y^{jq} \, .$$

The first equality holds as all q factors vanish and the second equality holds as $c_{ij} \in \mathbb{F}_q$ for every i and j. Namely, we have $p_E(x, y)^q = p_E(x^q, y^q)$ and therefore $P \in E(\overline{\mathbb{F}_q})$ if and only if $\Phi_q(P) \in E(\overline{\mathbb{F}_q})$.

We further have the three following properties for any positive integer k:

(i) for every $P \in E(\mathbb{F}_{q^k})$, $\Phi_q(P) \in E(\mathbb{F}_{q^k})$, hence Φ_q can be seen as a map $E(\mathbb{F}_{q^k}) \to E(\mathbb{F}_{q^k})$;

(ii) for every $P \in E(\overline{\mathbb{F}_q})$ we have $P \in E(\mathbb{F}_{q^k})$ if and only if $\Phi_q^k(P) = P$;

(iii) Φ_q^k is an group endomorphism and in particular $\Phi_q^k(P_1 + P_2) = \Phi_q^k(P_1) + \Phi_q^k(P_2)$ for every $P_1, P_2 \in E(\overline{\mathbb{F}_q})$

The first property follows from the same reasoning as above. The second property results from the fact that x lies in $\mathbb{F}_{q^k}$ if and only if $x^{q^k} = x$. The third property holds since every map from an elliptic curve to itself is a group endomorphism (see [Sil86]).

Definition 2.8 (trace map)**.** The *trace map* over $E(\mathbb{F}_{q^k})$ is the mapping:

$$\begin{aligned} \mathrm{Tr} \ : \ E(\mathbb{F}_{q^k}) &\longrightarrow E(\mathbb{F}_q) \\ P &\longmapsto P + \Phi_q(P) + \Phi_q^2(P) + \cdots + \Phi_q^k(P) \end{aligned}$$

Since the Φ_q^i's are morphisms, so is the trace map. Moreover, for every $P \in E(\mathbb{F}_{q^k})$, the trace map satisfies:

$$\mathrm{Tr}(\Phi_q(P)) = \Phi_q(\mathrm{Tr}(P)) = \mathrm{Tr}(P)$$

(which in particular shows that the range of the trace map is well $E(\mathbb{F}_q)$).

Let r be a prime co-prime to q and $q - 1$. The morphism structures of Φ_q and Tr imply that they can be seen as 2-dimensional linear applications over $E[r] \simeq \mathbb{Z}_r \times \mathbb{Z}_r$. It is then interesting to determine their eigenvalues and the corresponding eigenspaces.

The characteristic polynomial of the Frobenius is $\pi(x) = x^2 - tx + q$ where t is called *the trace of the Frobenius endomorphism* (which has been introduced in Section 2.1 as the trace of E). The Frobenius eigenvalues over $E[r]$ are then the roots of this polynomial modulo r, which are 1 and q since $\pi(x) \equiv (x - 1)(x - q) \bmod r$ (recalling that $r \mid q + 1 - t$).

Proposition 2.8. *The* 1*-eigenspace of the Frobenius over* $E[r]$ *is the subgroup of rational points* $E(\mathbb{F}_q)[r]$*. The* q*-eigenspace of the Frobenius over* $E[r]$ *is the subgroup generated by* $R - \Phi_q(R)$ *for every* $R \in E[r] \backslash E(\mathbb{F}_q)[r]$*.*

Proof. The first statement straightforwardly holds as $\Phi_q(P) = P$ for every $P \in E(\mathbb{F}_q)$. For the second statement, let Q be a q-eigenvector of Φ_q and let $P \in E(\mathbb{F}_q)[r]$. For every $R \in E[r]\backslash E(\mathbb{F}_q)[r]$, there exist $a, b \in \mathbb{Z}_r$ with $b \neq 0$ such that $R = [a]P+[b]Q$ and $R-\Phi_q(R) = [a]P+[b]Q-([a]P+[bq]Q) = [b(q-1)]Q$. As $r \nmid q-1$, we have $[b(q-1)]Q \neq \mathcal{O}$ and $\langle Q\rangle = \langle R - \Phi_q(R)\rangle$. □

Proposition 2.9. *The subgroup of rational points $E(\mathbb{F}_q)[r]$ is the k-eigenspace of the trace map. The q-eigenspace of the Frobenius over $E[r]$ is the 0-eigenspace of the trace map.*

Proof. The first statement straightforwardly holds as $\mathrm{Tr}(P) = [k]P$ for every $P \in E(\mathbb{F}_q)$. For the second statement, let Q be a q-eigenvector of Φ_q, we have $\mathrm{Tr}(Q) = Q + [q]Q + [q^2]Q + \cdots + [q^k]Q = [(q^k-1)/(q-1)]Q$. As $r \mid q^k - 1$ and $r \nmid q-1$ we have $r \mid (q^k-1)/(q-1)$ which implies $\mathrm{Tr}(Q) = \mathcal{O}$. □

Let $P \in E(\mathbb{F}_q)[r]\backslash\{\mathcal{O}\}$, let $R \in E[r]\backslash E(\mathbb{F}_q)[r]$ and let $Q = R - \Phi_q(R)$. The two above propositions show that (P, Q) is a basis of $E[r]$ and that under this basis, we have:

$$\Phi_q \equiv \begin{pmatrix} 1 & 0 \\ 0 & q \end{pmatrix} \quad \text{and} \quad \mathrm{Tr} \equiv \begin{pmatrix} k & 0 \\ 0 & 0 \end{pmatrix}$$

Eventually note that $\langle Q\rangle$ is usually called *the trace-zero subgroup* of $E[r]$.

2.7 Function field and divisors

Let E be an elliptic curve defined over $\mathbb{F}_q$ with defining polynomial p_E (*i.e.* $p_E(x, y) = x^3 + ax + b - y^2$). Let K be either an extension $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$ (possibly $\mathbb{F}_q$) or the algebraic closure $\overline{\mathbb{F}_q}$ of $\mathbb{F}_q$. The *function field* $K(E)$ is the field of *rational functions* $E(K) \to K$. Namely every $f \in K(E)$ can be expressed as $f = f_1/f_2$ where f_1 and f_2 are bivariate polynomials with coefficients in K (*i.e.* $f_1, f_2 \in K[x, y]$). Furthermore, for every $f = f_1/f_2$ and $g = g_1/g_2$ in $K(E)$ we have the equivalence relation: $f \sim g$ (meaning $f = g$ in $K(E)$) if and only if $f_1g_2 - g_1f_2 = hp_E$ for some $h \in K[x, y]$. This equivalence relation is consistent since for every $P \in E(K)$ we have $p_E(P) = 0$, and consequently, if $f \sim g$ then for every $P \in E(K)$, we have $f_1(P)g_2(P) - g_1(P)f_2(P) = 0$ that is $f(P) = g(P)$.

Definition 2.9. A *uniformizer* of E at $P \in E(K)$ is a generator of the ideal $\{h \in K(E);\ h(P) = 0\}$. For every $P \in E(K)$, E has a unique uniformizer at P up to constants in $\overline{\mathbb{F}_q}^*$. Let f be a non-zero function of $K(E)$. Then the *multiplicity* of f at P, denoted $\mathrm{ord}_P(f)$, is the unique integer n such that $f = gu^n$ where $g(P) \in K^*$ and u is a uniformizer of E at P. We have $f(P) \in K^*$ if and only if $\mathrm{ord}_P(f) = 0$. If $\mathrm{ord}_P(f) > 0$ (*i.e.* $f(P) = 0$) then f is said to have a *zero* at P and if $\mathrm{ord}_P(f) < 0$ (*i.e.* $f(P)$ is undefined/zero divides $f(P)$) then f is said to have a *pole* at P.

Definition 2.10 (divisor)**.** Let $(n_P)_{P\in E(\overline{\mathbb{F}_q})}$ be integers. A *divisor* D on E is a formal sum:

$$D = \sum_{P\in E\left(\overline{\mathbb{F}_q}\right)} n_P(P)$$

with finite *support* $\mathrm{supp}(D) = \{P;\ n_P \neq 0\}$. The set of divisors on E is denoted $\mathrm{Div}(E)$. It has a natural group structure, with addition law $\sum_P n_P(P) + \sum_P n'_P(P) = \sum_P (n_P + n'_P)(P)$.

Remark. The set of divisors on E can be thought as the additive group obtained from the points of E without any structure for the addition law. Every finite sum of points is a divisor and we have $\sum_P n_P(P) = \sum_P n'_P(P)$ if and only if $n_P = n'_P$ for every P. Such group could be actually defined from any set E without algebraic structure. However, as we will see in the following, the algebraic structure of elliptic curves makes divisors a very useful tool.

The *degree* of a divisor $D = \sum_P n_P(P)$ is defined as $\deg(D) = \sum_P n_P$. The set of divisor with degree 0, which is denoted $\mathrm{Div}^0(E)$, is a proper subgroup of $\mathrm{Div}(E)$.

Definition 2.11. The *divisor of the function* $f \in \overline{\mathbb{F}_q}(E)^*$, denoted $\mathrm{div}(f)$ is defined as:

$$\mathrm{div}(f) = \sum_{P \in E\left(\overline{\mathbb{F}_q}\right)} \mathrm{ord}_P(f)(P).$$

Two observations follow from this definition:

1. For every $f, g \in \overline{\mathbb{F}_q}(E)^*$, $\mathrm{div}(fg) = \mathrm{div}(f) + \mathrm{div}(g)$ and $\mathrm{div}(f/g) = \mathrm{div}(f) - \mathrm{div}(g)$.
2. If $\mathrm{div}(f) = \mathrm{div}(g)$ then $\mathrm{div}(f/g) = 0$ that is f/g is constant. We deduce that $\mathrm{div}(f)$ determines f up to constants in $\overline{\mathbb{F}_q}^*$.

Definition 2.12. A *principal divisor* is a divisor which equals $\mathrm{div}(f)$ for some function $f \in \overline{\mathbb{F}_q}(E)^*$.

Theorem 2.10 (Prop. II.3.1 [Sil86])**.** *Let E be an elliptic curve defined over $\mathbb{F}_q$. For every $f \in \overline{\mathbb{F}_q}(E)^*$, $\deg(\mathrm{div}(f)) = 0$.*

As a consequence of Theorem 2.10, the principal divisors form a subgroup of $\mathrm{Div}^0(E) \subset \mathrm{Div}(E)$.

Definition 2.13. Two divisors D and D' are said *equivalent*, denoted $D \sim D'$, if there exists $f \in \overline{\mathbb{F}_q}(E)^*$ such that $D = D' + \mathrm{div}(f)$.

In other words, the difference between two equivalent divisors is a principal divisor. Moreover, the equivalence classes of divisors form a group called the *divisor class group* (or *Picard group*) which is the quotient of $\mathrm{Div}(E)$ by the subgroup of principal divisors.

Theorem 2.11. *Let E be an elliptic curve defined over $\mathbb{F}_q$. Let $D = \sum_P n_P(P)$ be a degree 0 divisor on E. Then, there exists $f \in \overline{\mathbb{F}_q}(E)^*$ such that $D = \mathrm{div}(f)$ (and equivalently $D \sim 0$) if and only if $\sum_P [n_P]P = \mathcal{O}$.*

Definition 2.14. Let f be a function and let $D = \sum_P n_P(P)$ of degree 0 such that the supports of D and of $\mathrm{div}(f)$ are disjoint. We define:

$$f(D) = \prod_P f(P)^{n_P}.$$

Note that if $g = cf$ for some $c \in \overline{\mathbb{F}_q}^*$ then for every divisor D of degree 0 we have $f(D) = g(D)$. That is, $f(D)$ only depends on D and of $\mathrm{div}(f)$. Eventually, the following theorem is useful for defining Weil and Tate pairings (a proof is given in [Gal05, App.]).

Theorem 2.12 (Weil reciprocity law)**.** *Let E be an elliptic curve defined over $\mathbb{F}_q$. Let f and g be non-zero functions of $\overline{\mathbb{F}_q}(E)$ such that (f) and (g) have disjoint supports. Then:*

$$f\big(\mathrm{div}(g)\big) = g\big(\mathrm{div}(f)\big) \ .$$

3 Pairings

3.1 Basic definition

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two additive Abelian groups with identity element denoted $\mathcal{O}$ and let $\mathbb{G}_T$ be a multiplicative Abelian group with identity element denoted 1. In practice $\mathbb{G}_1$ and $\mathbb{G}_2$ will be some subgroups of points of an elliptic curve and $\mathbb{G}_T$ will be a subgroup of the multiplicative group of a finite field $\mathbb{F}_{q^k}$.

Definition 3.1. A *pairing* is a function:

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$$

satisfying the following properties:

(i) *bilinearity:* for every $P, P_1, P_2 \in \mathbb{G}_1$:

$$e(P, Q_1 + Q_2) = e(P, Q_1)\, e(P, Q_2)\ ,$$

and for every $Q, Q_1, Q_2 \in \mathbb{G}_2$:

$$e(P_1 + P_2, Q) = e(P_1, Q)\, e(P_2, Q)\ ,$$

(ii) *non-degeneracy:* if $e(P, Q) = 1$ for every $Q \in \mathbb{G}_2$, then $P = \mathcal{O}$ and if $e(P, Q) = 1$ for every $P \in \mathbb{G}_1$, then $Q = \mathcal{O}$,

(iii) *efficiency:* given any $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$, $e(P, Q)$ can be efficiently computed.

From the bilinearity property we can further deduce that, for every $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$, a pairing e satisfies $e(P, \mathcal{O}) = e(\mathcal{O}, Q) = 1$ and $e([a]P, [b]Q) = e(P, Q)^{ab} = e([b]P, [a]Q)$.

The bilinearity and non-degeneracy properties also imply that the greatest common divisor of the three group orders $\#\mathbb{G}_1$, $\#\mathbb{G}_2$ and $\#\mathbb{G}_T$ is greater than one. For cryptographic applications, one usually choose $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ to be cyclic groups of prime order r. In that case, one can notice that there exists essentially only one pairing: let e and e' be two different pairings $\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ and let $e(P, Q) = e_0$ and $e'(P, Q) = e'_0$ for some $(P, Q) \in \mathbb{G}_1 \times \mathbb{G}_2$. Then let $\alpha = \log_{e_0} e'_0$, that is $e'_0 = e_0^{\alpha}$. By bilinearity, $e'(\cdot, \cdot) = e(\cdot, \cdot)^{\alpha}$. We deduce that every pairing $\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a power of e, in other words every pairing is an element of the multiplicative group of order r generated by e.

While using a pairing to design a cryptographic protocol, three different settings appear depending on the existence or not of *efficiently computable* isomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$ [GPS06]. Note that isomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$ always exist but they are not necessarily efficiently computable (*e.g.* $P' \mapsto [\log_P P']Q$ where $P \in \mathbb{G}_1^*$ and $Q \in \mathbb{G}_2^*$). We then consider three different types of pairings:

– **Type I:** There exist efficiently computable isomorphisms $\phi : \mathbb{G}_1 \to \mathbb{G}_2$ and $\psi : \mathbb{G}_2 \to \mathbb{G}_1$. This type of pairing is called a *symmetric pairing* since it is virtually equivalent to taking $\mathbb{G}_1 = \mathbb{G}_2$. Indeed if $\mathbb{G}_1 = \mathbb{G}_2$ then ϕ and

ψ exist (with $\phi = \psi = \mathrm{Id}_{\mathbb{G}_1}$ as a particular case), and if ϕ and ψ exist then one can define a symmetric pairing over $\mathbb{G}_1 \times \mathbb{G}_1$ as $\hat{e}(\cdot,\cdot) = e(\cdot, \psi(\cdot))$ (the bilinearity and the non-degeneracy of e implies that of $\hat{e}$ as ψ is an isomorphism).

- **Type II:** There exists an efficiently computable isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ but there exist no efficiently computable isomorphism $\phi : \mathbb{G}_1 \to \mathbb{G}_2$ (and in particular $\mathbb{G}_1 \neq \mathbb{G}_2$ otherwise $\phi = \mathrm{Id}_{\mathbb{G}_1}$ would exist).
- **Type III:** There exist no efficiently computable isomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$ (and in particular $\mathbb{G}_1 \neq \mathbb{G}_2$).

In contrast to symmetric pairings (type I), pairings of types II and III are called *asymmetric pairings*. Note that a symmetric pairing $e(\cdot, \psi(\cdot))$ can always be constructed from an asymmetric pairing of type II.

In the following we will describe two pairings which can be used to construct type I, II or III cryptographic pairings: the Weil pairing and the Tate pairing.

3.2 The Weil Pairing

Let E be an elliptic curve defined over $\mathbb{F}_q$ and let $r | \#E(\mathbb{F}_q)$ co-prime to q. Let k be the embedding degree of $E(\mathbb{F}_q)$ with respect to r and assume $k > 1$ (*i.e.* $r \nmid q - 1$). According to Proposition 2.4, we have $E[r] = E(\mathbb{F}_{q^k})[r]$. The *Weil pairing* is defined as:

$$\begin{aligned} w_r \ : \ E[r] \times E[r] &\longrightarrow \mu_r \subset \mathbb{F}_{q^k}^* \\ (P, Q) &\longmapsto f_P(D_Q)/f_Q(D_P) \end{aligned}$$

where D_P and D_Q are divisors on E with $D_P \sim (P) - (\mathcal{O})$, $D_Q \sim (Q) - (\mathcal{O})$ and $\mathrm{supp}(D_P) \cap \mathrm{supp}(D_Q) = \emptyset$, and where f_P and f_Q are functions of $\mathbb{F}_{q^k}(E)$ with $\mathrm{div}(f_P) = rD_P$ and $\mathrm{div}(f_Q) = rD_Q$ (such functions exist according to Theorem 2.11).

Consistency. Let us first show that $f_P(D_Q)/f_Q(D_P) \in \mu_r$. For every $P, Q \in E[r]$, we have:

$$\left(\frac{f_P(D_Q)}{f_Q(D_P)}\right)^r = \frac{f_P(rD_Q)}{f_Q(rD_P)} = \frac{f_P(\mathrm{div}(f_Q))}{f_Q(\mathrm{div}(f_P))} = 1,$$

where the last equality holds by the Weil reciprocity (see Theorem 2.12). We now show that the Weil pairing is consistent for every choice of D_P, D_Q, f_P and f_Q satisfying the aforementioned properties. On the one hand, if f_P and f'_P are such that $\mathrm{div}(f_P) = \mathrm{div}(f'_P)$ then there exists $c \in \mathbb{F}_{q^k}^*$ such that $f'_P = cf_P$ and:

$$f'_P(D_Q) = (cf_P)(D_Q) = f_P(D_Q)$$

(the same argument holds for f_Q). On the other hand, choosing D'_P instead of D_P implies computing $f'_P(D_Q)/f_Q(D'_P)$ where $\mathrm{div}(f'_P) = rD'_P$. If $D'_P \sim D_P$, then there exists a function $g \in \mathbb{F}_{q^k}(E)^*$ such that $D'_P = D_P + \mathrm{div}(g)$ and $f'_P = f_P g^r$, which implies:

$$\frac{f'_P(D_Q)}{f_Q(D'_P)} = \frac{f_P(D_Q)g^r(D_Q)}{f_Q(D_P)f_Q(\mathrm{div}(g))} = \frac{f_P(D_Q)g^r(D_Q)}{f_Q(D_P)g(\mathrm{div}(f_Q))} = \frac{f_P(D_Q)}{f_Q(D_P)}.$$

Note that the equality $f_Q\big(\mathrm{div}(g)\big) = g\big(\mathrm{div}(f_Q)\big)$ is due to the Weil reciprocity (see Theorem 2.12). The same reasoning holds for the choice of D_Q which shows the consistency of the Weil pairing.

Theorem 3.1 (Weil pairing properties)**.** *The Weil pairing satisfies the bilinearity and the non-degeneracy properties. The Weil pairing further satisfies the following* alternating property*: for every $P, Q \in E[r]$ we have $w_r(P,Q) = e(Q,P)^{-1}$, and in particular $w_r(P,P) = 1$.*

Proof. The alternating property straightforwardly results by definition of the Weil pairing. For the proof of non-degeneracy see [Gal05]. The proof of bilinearity is quite similar to the one for the Tate pairing (see proof of Theorem 3.2 hereafter). In particular it is shown that $f_{P_1+P_2}(D_Q) = f_{P_1}(D_Q) f_{P_2}(D_Q)\alpha_r$ and $f_Q(D_{P_1} + D_{P_2}) = f_Q(D_{P_1}) f_Q(D_{P_2})\beta_r$ for some $\alpha, \beta \in \mathbb{F}_{q^k}^*$. Therefore we have $w_r(P_1 + P_2, Q) = w_r(P_1, Q) w_r(P_2, Q)(\alpha\beta^{-1})^r$ and since w_r values lies in μ_r, we deduce $(\alpha\beta^{-1})^r \in \mu_r$ that is $(\alpha\beta^{-1})^r = 1$. □

A consequence of the alternating property of the Weil pairing (and in particular of the fact that $w_r(P,P) = 1$ for every $P \in E[r]$) together with the bilinearity property is that for every $Q \in \langle P \rangle$ we have $w_r(P,Q) = 1$.

3.3 The Tate Pairing

Let π denote the canonical projection $\mathbb{F}_{q^k}^* \to \mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^r$. Namely, for every $\alpha, \beta \in \mathbb{F}_{q^k}^*$, we have $\pi(\alpha) = \pi(\beta)$ if and only if $\alpha = \beta\lambda^r$ for some $\lambda \in \mathbb{F}_{q^k}^*$.

Let E be an elliptic curve defined over $\mathbb{F}_q$ and let $r | \#E(\mathbb{F}_q)$ co-prime to q. Let k be the embedding degree of $E(\mathbb{F}_q)$ with respect to r and assume $k > 1$ (*i.e.* $r \nmid q - 1$). According to Proposition 2.4, we have $E[r] = E(\mathbb{F}_{q^k})[r]$. The *Tate pairing* is defined as:

$$\begin{aligned} \langle \cdot, \cdot \rangle_r \ : \ E[r] \times E(\mathbb{F}_{q^k}) &\longrightarrow \mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^r \\ (P, Q) &\longmapsto \langle P, Q \rangle_r = \pi(f_P(D_Q)) \end{aligned}$$

where f_P is any function of $\mathbb{F}_{q^k}(E)$ with $\mathrm{div}(f_P) = r(P) - r(\mathcal{O})$ and D_Q is any divisor on E equivalent to $(Q) - (\mathcal{O})$ with support disjoint from $\{\mathcal{O}, P\}$.

Consistency. We show hereafter that the Tate pairing is consistent for every choice of f_P and D_Q satisfying the aforementioned properties. As for the Weil pairing, changing f_P without modifying its divisor amounts to multiplying it by a constant in $\mathbb{F}_{q^k}^*$, which does not affect the value of $f_P(D_Q)$. On the other hand, if D_Q and D'_Q are such that $D_Q \sim D'_Q$ (but possibly $D_Q \neq D'_Q$), then we may have $f_P(D_Q) \neq f_P(D'_Q)$ but $\pi(f_P(D_Q)) = \pi(f'_P(D'_Q))$ always holds (which explains why the values of the Tate pairing are equivalence classes). Indeed, if $D'_Q \sim D_Q$ then there exists $g \in \mathbb{F}_{q^k}(E)$ such that $D'_Q = D_Q + \mathrm{div}(g)$ and:

$$f_P(D'_Q) = f_P(D_Q) f_P\big(\mathrm{div}(g)\big) = f_P(D_Q) g\big(\mathrm{div}(f_P)\big) = f_P(D_Q) \left(\frac{g(P)}{g(\mathcal{O})}\right)^r ,$$

that is $\pi\big(f_P(D'_Q)\big) = \pi\big(f_P(D_Q)\big)$. Note that the second equality is due to the Weil reciprocity (see Theorem 2.12).

The reduced Tate Pairing. In practice, it is more convenient to work with actual values in $\mathbb{F}^*_{q^k}$ rather than with equivalence classes. The exponentiation to the $(q^k - 1)/r$ maps every class of equivalence in $\mathbb{F}^*_{q^k}/(\mathbb{F}^*_{q^k})^r$ to a single element in $\mu_r \subset (\mathbb{F}_{q^k})^*$. The *reduced Tate pairing* is then defined as:

$$\begin{aligned} t_r \ : \ E[r] \times E(\mathbb{F}_{q^k}) &\longrightarrow \mu_r \subset \mathbb{F}^*_{q^k} \\ (P, Q) &\longmapsto \langle P, Q\rangle_r^{(q^k-1)/r} \end{aligned}$$

Using the previous notations we have $t_r(P, Q) = f_P(D_Q)^{(q^k-1)/r}$.

This definition of the Tate pairing is equivalent of the one given above in the sense that the exponentiation to the $(q^k - 1)/r$ is an isomorphism from $\mathbb{F}^*_{q^k}/(\mathbb{F}^*_{q^k})^r$ to $\mu_r \subset \mathbb{F}^*_{q^k}$.

Theorem 3.2 (Tate pairing properties)**.** *The Tate pairing satisfies the bilinearity and the non-degeneracy properties.*

Proof. For a proof of the non-degeneracy see [Gal05], we prove hereafter the bilinearity. For every $Q_1, Q_2 \in E(K)$, we have $f_P(D_{Q_1+Q_2}) = f_P(D_{Q_1} + D_{Q_2}) = f_P(D_{Q_1}) f_P(D_{Q_2})$ for every $P \in E[r]$. On the other hand, for every $P_1, P_2 \in E[r]$ we have $\mathrm{div}(f_{P_1+P_2}) = \mathrm{div}(g^r f_{P_1} f_{P_2})$ where $g \in \mathbb{F}_{q^k}(E)$ is such that $\mathrm{div}(g) = (P_1 + P_2) - (P_1) - (P_2) + (\mathcal{O})$, which implies $\pi(f_{P_1+P_2}(D_Q)) = \pi(f_{P_1}(D_Q) f_{P_2}(D_Q))$ for every $Q \in E(\mathbb{F}_{q^k})$. □

Let $P \in E[r]$ and $Q, Q' \in E(\mathbb{F}_{q^k})$ such that $Q' = Q + [r]R$ for some $R \in E(\mathbb{F}_{q^k})$. We have:

$$t_r(P, Q') = t_r(P, Q) t_r(P, R)^r = t_r(P, Q) \ .$$

In other words, two points Q and Q' which lie in the same equivalence class of $E(\mathbb{F}_{q^k})/[r]E(\mathbb{F}_{q^k})$ are such that $t_r(P, Q') = t_r(P, Q)$ for every P. For this reason the Tate pairing is sometimes defined as a map $E[r] \times E(\mathbb{F}_{q^k})/[r]E(\mathbb{F}_{q^k}) \to \mu_r \subset \mathbb{F}^*_{q^k}$. However, in practice it is more convenient to work with a set of points representing $E(\mathbb{F}_{q^k})/[r]E(\mathbb{F}_{q^k})$ *i.e.* for which every element lies in a distinct class.

Lemma 3.3. *If* $r^3 \nmid \#E(\mathbb{F}_{q^k})$ *then* $E[r] \cap [r]E(\mathbb{F}_{q^k}) = \{\mathcal{O}\}$.

The above lemma states that when $r^3 \nmid \#E(\mathbb{F}_{q^k})$ (which usually occurs in practice), no r-torsion point of E lies in $[r]E(\mathbb{F}_{q^k})$ except the identity $\mathcal{O}$. In other words, every points in $E[r]$ corresponds to a distinct class of $E(\mathbb{F}_{q^k})/[r]E(\mathbb{F}_{q^k})$. In that case, the Tate pairing can be defined as a map:

$$t_r \ : \ E[r] \times E[r] \longrightarrow \mu_r \subset \mathbb{F}^*_{q^k} \ .$$

In the following we will assume $r^3 \nmid \#E(\mathbb{F}_{q^k})$ and we will consider $E[r] \times E[r]$ to be the domain of the Tate pairing.

Finally we have the next degeneracy result for the restriction of the Tate pairing over $E(\mathbb{F}_q)$.

Proposition 3.4 (Lem. IX.8 [Gal05]). *Let E be an elliptic curve defined over $\mathbb{F}_q$, let $r|\#E(\mathbb{F}_q)$ such that $\gcd(r,q)=1$, and let $k>1$ be the embedding degree of $E(\mathbb{F}_q)$ with respect to r. Then for every $P\in E(\mathbb{F}_q)[r]$ and $Q\in E(\mathbb{F}_q)$, $\langle P,Q\rangle_r = (\mathbb{F}_{q^k}^*)^r$ and $t_r(P,Q)=1$.*

3.4 Pairings over cyclic subgroups

We discuss here the selection of the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ for the definition of the pairing $e:\mathbb{G}_1\times\mathbb{G}_2\to\mathbb{G}_T$. Let us first note that, except in some specific applications requiring composite order groups, r is always chosen to be a prime for security reasons. Indeed, the security of all pairing-based protocols requires that the discrete logarithm problem is hard over $E[r]$ and $\mu_r\subset\mathbb{F}_{q^k}^*$. The best choice to satisfy this requirement is to take r as a (large) prime. In that case $E[r]$ is isomorphic to $\mathbb{Z}_r\times\mathbb{Z}_r$ that is it contains r cyclic subgroups of order r and every $P\in E[r]\backslash\{\mathcal{O}\}$ belongs to one and only one of these subgroups. The groups $\mathbb{G}_1$ and $\mathbb{G}_2$ are then chosen as proper subgroups of $E[r]$ and there exist $P,Q\in E[r]$ such that $\mathbb{G}_1=\langle P\rangle$ and $\mathbb{G}_2=\langle Q\rangle$.

The most natural choice for $\mathbb{G}_1$ is to take the subgroup of rational points $E(\mathbb{F}_q)[r]$. Indeed, storing a point of $E(\mathbb{F}_{q^k})$ requires k times more memory than storing a point of $E(\mathbb{F}_q)$ and computing a point addition over $E(\mathbb{F}_{q^k})$ is expected to take k^2 times the time of the same operation over $E(\mathbb{F}_q)$. For these reasons, working with rational points is always preferred to working with $\mathbb{F}_{q^k}$-rational points, and in practice one uses $\mathbb{G}_1=E(\mathbb{F}_q)[r]$.

On the other hand, we have seen that for both Weil and Tate pairings we have $e(P,P)=1$ for every $P\in E(\mathbb{F}_q)[r]$ (for the Weil pairing, this is further true for any $P\in E[r]$), hence choosing $\mathbb{G}_2=\mathbb{G}_1=E(\mathbb{F}_q)[r]$ will yield a degenerate pairing. Therefore, to get a non-degenerate pairing, one has to choose $\mathbb{G}_2$ as a distinct subgroup of $E[r]$ *i.e.* as one of the $r-1$ non-rational subgroups. The choice of this subgroup then depends on the type of pairing (I, II or III–see Section 3.1) which is desired.

Type I. In order to construct a symmetric pairing while $\mathbb{G}_1=E(\mathbb{F}_q)[r]$, one needs an efficiently computable isomorphism from the rational subgroup of $E[r]$ to a non-rational subgroup of $E[r]$. The only known way to construct such an isomorphism is by using a distortion map (*i.e.* an endomorphism of the curve mapping rational points to non-rational points in $E(\mathbb{F}_{q^k})$). As explained in Section 2.4, distortion maps only exist over supersingular elliptic curves. Therefore, defining a symmetric pairing (with $\mathbb{G}_1=E(\mathbb{F}_q)[r]$) requires the use of a supersingular elliptic curve, which restricts the choice of the curve to those defined in Table 2 (see Section 2.4). Then $\mathbb{G}_2$ is defined as $\mathbb{G}_2=\langle\phi(P)\rangle$ for any $P\in\mathbb{G}_1$ and the trace map is an (efficiently computable) isomorphism $\mathbb{G}_2\to\mathbb{G}_1$, which completes the requirements for the symmetric pairing.

Remark. Another possibility is to take both $\mathbb{G}_1$ and $\mathbb{G}_2$ among the non-rational subgroups. For instance, one can select a non-rational point $P\in E(\mathbb{F}_{q^k})[r]$ and set $\mathbb{G}_1=\langle P\rangle$ and $\mathbb{G}_2=\langle\phi(P)\rangle$ where ϕ is the q^d-power Frobenius for some

$d < k$. However such an approach is unlikely to be followed in practice as it would yield a quite inefficient pairing.

Type II. An asymmetric pairing of type II can be constructed using any ordinary elliptic curve by taking $\mathbb{G}_2$ as a non-rational subgroup of $E[r]$ (still with $\mathbb{G}_1 = E(\mathbb{F}_q)[r]$). In opposition to the previous case, this ensures that there exist *a priori* no efficiently computable isomorphism from $\mathbb{G}_1$ to $\mathbb{G}_2$. An additional requirement is that $\mathbb{G}_2$ is not the trace-zero subgroup so that the trace is an (efficiently computable) isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$ (see Section 2.6). To summarize, $\mathbb{G}_2$ is chosen as one of the $r - 2$ subgroups of order r which is neither the rational subgroup nor the trace-zero subgroup.

Type III. Finally, an asymmetric pairing of type III is constructed using any ordinary elliptic curve by taking $\mathbb{G}_1$ as the rational subgroup and $\mathbb{G}_2$ as the trace-zero subgroup. As we will explain in Section 5, such type III pairings are the most efficient to compute.

3.5 Pairing-friendly elliptic curves

All elliptic curves are not suitable for the computation of Weil and Tate pairings. In fact, most ordinary elliptic curves have embedding degrees k which are about the same size as q which renders the arithmetic over $\mathbb{F}_{q^k}$ clearly impractical. A *pairing-friendly elliptic curve* is an elliptic curve E defined over $\mathbb{F}_q$ such that $\#E(\mathbb{F}_q)$ admits a large prime factor r and the embedding degree k of E with respect to r is small (typically smaller that 30). As we will see in Section 4.1, the value of the embedding degree must be actually chosen accordingly to the desired security level for the underlying cryptographic protocol.

Many families of pairing-friendly elliptic curves have been proposed in the literature and giving an exhaustive list is out of the scope of this document. We refer the reader to [FST10] which gives a complete and up-to-date taxonomy of pairing-friendly elliptic curves.

4 Promising pairing-based protocols

In this section, we present three cryptographic protocols which make use of pairings to achieve specific properties. The security of these protocols are based on certain hardness assumptions which are recalled hereafter.

4.1 Hardness assumptions

The security of cryptographic protocols is based on some *hardness assumptions* which state that some mathematical problems are hard to solve (meaning that they cannot be solved in a reasonable time with a reasonable computational resource). Under such assumptions, cryptographic protocols can be formally proved to be secure: one shows that if the system can be efficiently broken then the underlying problem can be efficiently solved, which by assumption is impossible. We present hereafter some of these problems which are used to prove the security of elliptic curve and pairing-based protocols.

Let $\mathbb{G}$ denote an Abelian group of order r. The three following problems are widely used in cryptography.

- **Discrete Logarithm (DL):** Given $P \in \mathbb{G}$ and $[a]P$, compute a.
- **Computational Diffie-Hellman (CDH):** Given $P \in \mathbb{G}$, $[a]P$, and $[b]P$, compute $[ab]P$.
- **Decisional Diffie-Hellman (DDH):** Given $P \in \mathbb{G}$, $[a]P$, $[b]P$, and $[c]P$, decide whether $ab \equiv c \bmod r$.

The DL problem is believed to be hard for carefully chosen groups of large order (we will see later what *large* means) including the multiplicative group of a finite field and the group of points of an elliptic curve defined over a finite field.

It is simple to see that if one can solve the DL problem efficiently then one can also solve CDH and DDH efficiently. Also, the ability of solving CDH implies that of solving DDH. It is further known that CDH is almost as hard as DL over many groups [BL96, dB88, MW99]. This can be summarized with the following inequality between the problem difficulties:

$$\mathrm{DL} \simeq \mathrm{CDH} \geq \mathrm{DDH}\ .$$

On the other hand, there exist groups for which DDH is easy while CDH is (presumably) hard *i.e.* for which:

$$\mathrm{DL} \simeq \mathrm{CDH} > \mathrm{DDH} = easy.$$

These groups, called *gap Diffie-Hellman groups*, are typically groups $\mathbb{G}$ for which a symmetric pairing $\hat{e} : \mathbb{G}\times\mathbb{G} \to \mathbb{G}_T$ exists. On these groups, we have $\hat{e}([a]P, [b]P) = \hat{e}(P, [c]P)$ if and only if $ab \equiv c \bmod r$ (by bilinearity of the pairing), which provides a simple way to solve DDH although CDH is still (presumably) hard.

The definition of gap Diffie-Hellman groups can be extended to pair of groups $(\mathbb{G}_1, \mathbb{G}_2)$ for which there exists an asymmetric pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Let r be the order of $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$. The following problems generalize CDH and DDH:

- **Computational Co-Diffie-Hellman (Co-CDH):** Given $Q \in \mathbb{G}_2$, $[a]Q$ and $P \in \mathbb{G}_1$, compute $[a]P$.
- **Decisional Co-Diffie-Hellman (Co-DDH):** Given $P \in \mathbb{G}_1$, $[a]P$, $Q \in \mathbb{G}_2$ and $[b]Q$, decide whether $a \equiv b \bmod r$.

When $\mathbb{G}_1 = \mathbb{G}_2$ these problems are equivalent to the standard CDH and DDH. When $\mathbb{G}_1 \neq \mathbb{G}_2$, the existence of an asymmetric pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ implies that co-DDH is easy (by testing whether $e(P, [b]Q) = e([a]P, Q)$) while co-CDH is still (presumably) hard. The pair $(\mathbb{G}_1, \mathbb{G}_2)$ is then called a *gap co-Diffie-Hellman pair*.

To summarize, the existence of a pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ always enables to solve co-DDH over $(\mathbb{G}_1, \mathbb{G}_2)$ while solving DDH over $\mathbb{G}$ requires the existence of a symmetric pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. When CDH and co-CDH are hard, the existence of a pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ implies:

(i) $(\mathbb{G}_1, \mathbb{G}_2)$ is a co-gap Diffie-Hellman pair;

(ii) $\mathbb{G}_1$ is a gap Diffie-Hellman group only in the type I setting;

(iii) $\mathbb{G}_2$ is a gap Diffie-Hellman group only in the type I and type II settings.

Note that in groups which are not gap Diffie-Hellman, the only known way to solve DDH is to compute $[ab]P$ and then check whether $[ab]P = [c]P$. DDH is then considered to be as hard as CDH (and hence almost as hard as DL).

In order to prove the security of pairing-based protocols, a further generalization of CDH has been introduced. Let $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a symmetric pairing and let consider the following problem.

- **Bilinear Diffie-Hellman (BDH):** Given $P \in \mathbb{G}$, $[a]P$, $[b]P$ and $[c]P$, compute $\hat{e}(P, P)^{abc}$.

It is simple to see that the ability of solving CDH in either $\mathbb{G}$ or $\mathbb{G}_T$ implies the ability of solving BDH. Nothing else is known about the hardness of BDH but it is usually assumed to be as hard as the easier among CDH in $\mathbb{G}$ and CDH in $\mathbb{G}_T$.

Another important hardness relation due to pairing is the following. The DL problem over groups $\mathbb{G}_1$ and $\mathbb{G}_2$ such that there exists a pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is not harder than the DL problem over $\mathbb{G}_T$. Indeed, as $e([a]P, Q) = e(P, Q)^a$, the solution of the DL instance $(P, [a]P)$ over $\mathbb{G}_1$ is also the solution of the DL instance $(e(P, Q), e(P, Q)^a)$ over $\mathbb{G}_T$ (the same reasoning applies for $\mathbb{G}_2$). This relation has been originally used as an "attack" against elliptic curve cryptosystems [MOV91, MOV93, FR94]. Let E be an elliptic curve over $\mathbb{F}_q$ and suppose that the DL problem over $E(\mathbb{F}_q)[r]$ is used to design some cryptographic protocol for some large $r \mid \#E(\mathbb{F}_q)$. Let k be the embedding degree of E with respect to r. The Weil and Tate pairings enable to transport the DL problem over $E(\mathbb{F}_q)[r]$ to the DL problem over $\mu_r \subset \mathbb{F}^*_{q^k}$. If the embedding degree k is low (as for instance for supersingular elliptic curves–see Section 2.4), then it might be easier to solve the DL problem in $\mathbb{F}^*_{q^k}$ than to solve it in $E(\mathbb{F}_q)$. Indeed, we only know exponential-time algorithms to solve DL over elliptic curves whereas we know subexponential-time algorithms to solve DL over multiplicative groups of finite fields. Consequently, while using the Weil and the Tate pairing to implement some

Table 3: Minimal bit-size of r and q^k for the DL problem over $E(\mathbb{F}_q)[r]$ and $\mathbb{F}^*_{q^k}$ to be hard w.r.t. a given security level.

security bits	$\lvert r\rvert$ for DL in $E(\mathbb{F}_q)$	$\lvert q^k\rvert$ for DL in $\mathbb{F}^*_{q^k}$
80	160	1248
128	256	3248
256	512	15424

cryptographic protocol, one must take the group $\mathbb{F}^*_{q^k}$ substantially larger than r in order to ensure the same difficulty level for the DL in both groups.

Table 3 gives the minimal bit-size of $r \mid \#E(\mathbb{F}_q)$ as well as the minimal bit-size of q^k to get a certain security level for the DL problem over $E(\mathbb{F}_q)[r]$ and $\mathbb{F}^*_{q^k}$ respectively (where r must be a prime). This level is given in terms of *security bits*: m security bits mean that the problem requires at least 2^m elementary operations to be solved. For a chosen security level, one must ensure that the sizes of r and q^k verify the lower bounds specified in Table 3.

4.2 Boneh-Franklin identity-based encryption scheme

Identity-based encryption (IBE) is a form of public-key encryption with the specification that a user's public key is simply his *identity* (or more generally any string that can be derived from public data identifying the user such as his name, his e-mail address, etc.). The main motivation of IBE is to solve the public key distribution issue of classical public-key cryptography. Indeed, with an IBE scheme Bob does not need to get Alice public key certificate to send her a ciphered message since he already knows Alice identity. On the other hand, Alice must obtain the private key corresponding to her identity from an authority called *Private Key Generator* (PKG).

Although the concept of IBE was introduced by Shamir in 1984 [Sha84], no actual solution was known at that time and the design of an IBE scheme remained an open problem until 2001 when Boneh and Franklin proposed the first efficient IBE scheme based on the use of pairings [BF01, BF03]. In Boneh-Franklin IBE scheme, the PKG selects (i) two groups $\mathbb{G} = \langle P \rangle$ and $\mathbb{G}_T$ of order r for which a symmetric pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ exists, (ii) two cryptographic hash functions $H_1 : \{0,1\}^* \to \mathbb{G}^*$ and $H_2 : \mathbb{G}_T \to \{0,1\}^n$, (iii) a random $s \in \mathbb{Z}_r$. The PKG set $P_{pub} = [s]P$ as its own public key and it publishes the scheme parameters:

$$(\mathbb{G}, \mathbb{G}_T, \hat{e}, H_1, H_2, P, P_{pub}) \ .$$

The message space is $\mathcal{M} = \{0,1\}^n$ and the PKG secret key, also called the *master key*, is s. The private key K_{ID} corresponding to a given identity $\mathsf{ID} \in \{0,1\}^*$ is defined as:

$$K_{\mathsf{ID}} = [s]Q_{\mathsf{ID}} \ ,$$

where $Q_{\mathsf{ID}} = H_1(\mathsf{ID}) \in \mathbb{G}^*$. Note that only the PKG is able to derive K_{ID} as

only the PKG knows the master key s (for everyone else, recovering s requires the ability of solving the DL problem).

The encryption of a message $m \in \mathcal{M}$ under a public key ID works as follows:

1. compute $Q_{\mathsf{ID}} = H_1(\mathsf{ID})$,
2. choose a random $k \in \mathbb{Z}_r$,
3. set the ciphertext to be:

$$c = \big([k]P, m \oplus H_2(g_{\mathsf{ID}}^k)\big) \quad \text{where} \quad g_{\mathsf{ID}} = \hat{e}(Q_{\mathsf{ID}}, P_{pub}) \in \mathbb{G}_T^* \,.$$

Then, the decryption of the ciphertext c from the private key K_{ID} works as follows:

1. compute $g_{\mathsf{ID}}^k = \hat{e}(K_{\mathsf{ID}}, [k]P)$. The latter equality indeed holds as:

$$g_{\mathsf{ID}}^k = \hat{e}(Q_{\mathsf{ID}}, P_{pub})^k = \hat{e}(Q_{\mathsf{ID}}, P)^{ks} = \hat{e}([s]Q_{\mathsf{ID}}, [k]P) = \hat{e}(K_{\mathsf{ID}}, [k]P) \,,$$

2. recover m by $m = \big(m \oplus H_2(g_{\mathsf{ID}}^k)\big) \oplus H_2(g_{\mathsf{ID}}^k)$.

The computational requirements of the Boneh-Franklin scheme are [2]:

– for encryption: one scalar multiplication, one hashing to $\mathbb{G}$ and one pairing evaluation,
– for decryption: one pairing evaluation with exponentiation of the result.

One can check that the security of the Boneh-Franklin scheme is related to the BDH problem. Denoting $Q_{\mathsf{ID}} = [a]P$ (and recalling $P_{pub} = [s]P$), the decryption of a message requires the computation of $g_{\mathsf{ID}}^k = \hat{e}(P, P)^{ask}$ from P, $[a]P$, $[s]P$ and $[k]P$, which is precisely a BDH instance. In [BF01], it is formally proved that the above scheme is secure under the assumption that BDH is a hard problem and that the hash functions H_1 and H_2 behave as *random oracles*. A further scheme is also described which is very close to the basic scheme presented above but which is proved to be secure under a *chosen ciphertext attack* scenario.

4.3 Boneh-Lynn-Shacham short signature scheme

For standard digital signature schemes, the size of the signature is at least four times the security level (typically for DSA and ECDSA) and it may be substantially larger (*e.g.* for RSA-based signature schemes such as RSA-PSS). For instance, to get a 80-bit security level, (EC)DSA signatures require to be 320 bits long while RSA-PSS signatures require to be 1248 bits long. For some applications, for instance where a signature shall be keyed in by a human, it would be more convenient to have shorter signatures. This was the motivation of the short signature scheme introduced by Boneh, Lynn and Shacham in 2001 which provides signatures whose size is only twice the security level [BLS01, BLS04]. They achieve this property by the use of a pairing.

The Boneh-Lynn-Shacham (BLS) signature scheme makes use of a co-gap Diffie-Hellman pair of groups $(\mathbb{G}_1, \mathbb{G}_2)$ of order r and for which an efficiently computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$ exists (this isomorphism is not used in the design but it is necessary to prove the security of the scheme). Such a pair $(\mathbb{G}_1, \mathbb{G}_2)$ is typically a pair of groups for which there exists a type II asymmetric

2. We omit the hashing to $\{0,1\}^n$ and the XOR which are basic and very efficient operations.

pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ (the pairing yields the co-gap Diffie-Hellman property and the type II yields the isomorphism $\mathbb{G}_2 \to \mathbb{G}_1$). The BLS signature scheme further makes use of a cryptographic hash function $H : \{0,1\}^* \to \mathbb{G}_1$.

Let Q be a public generator of $\mathbb{G}_2$. The generation of a public-private key pair consists in choosing a random $k \in \mathbb{Z}_r$ and setting $Q_{pub} = [k]Q$ as public key and k as private key.

The signature s of a message $m \in \{0,1\}^*$ under the private key k is simply given by $S = [k]P_m$ where $P_m = H(m)$. The verification of the signature consists in computing $P_m = H(m)$ and checking that (P_m, S, Q, Q_{pub}) is a valid co-DDH instance. If the signature is correct, we indeed have $(P_m, S, Q, Q_{pub}) = (P_m, [k]P_m, Q, [k]Q)$ which is a valid co-DDH instance. In practice this verification is done by checking whether $e(P_m, Q_{pub})$ equals $e(Q, S)$, and if the signature is valid we indeed have:

$$e(P_m, Q_{pub}) = e(P_m, Q)^k = e(S, Q) \; .$$

The computational requirements of the BLS scheme are:

– for signing: one hashing to $\mathbb{G}_1$ and one scalar multiplication of the hash value,
– for verifying: one hashing to $\mathbb{G}_1$ and two pairing evaluations.

To forge a valid signature for a message m while ignoring the private key k, one must compute $S = [k]P_m$ from P_m, Q and $Q_{pub} = [k]Q$, which is precisely a co-CDH instance. In [BLS01], it is formally proved that the BLS scheme is secure against *signature forgery* under the assumption that co-CDH over $(\mathbb{G}_1, \mathbb{G}_2)$ is a hard problem and that the hash function H behaves as a random oracle.

The shortness of the signature comes at the cost of a little modification of the scheme. The signature is no more defined as $S = [k]P_m = (x_S, y_S)$ but as x_S only: the y-coordinate of S is removed. Note that the x-coordinate of a point S is also the x-coordinate of its inverse $-S = (x_S, -y_S)$. The verification starts by recovering $y'_S \in \{y_S, -y_S\}$ from x_S. Then the signature is accepted as valid if and only if:

$$e(P_m, Q_{pub}) = e(S', Q) \quad \text{or} \quad e(P_m, Q_{pub}) = e(S', Q)^{-1},$$

where $S' = (x_S, y'_S) \in \{S, -S\}$. Note that the soundness of the verification holds from $e(S', Q)^{-1} = e(-S, Q)$. Therefore, compared to the previous scheme, $[k]P_m$ and $-[k]P_m$ are both valid signatures for a message m. As argued in [BLS01], this modification does not affect the security of the scheme. The verification still requires two pairing computations and it further requires an inversion in $\mathbb{G}_T$.

As explained in Section 3.4, a type II pairing is obtained by taking $\mathbb{G}_1 = E(\mathbb{F}_q)[r]$ and $\mathbb{G}_2$ as the trace-zero subgroup of $E(\mathbb{F}_{q^k})[r]$ (where $r \mid \#E(\mathbb{F}_q)$ and k is the embedding degree of E w.r.t. r). Note that the x-coordinate of a point in $\mathbb{G}_1$ then lies in $\mathbb{F}_q$ and therefore has the same size as q. Since to get an m-bit security level one must take $|q| \geq |r| \approx 2m$ for the hardness of the DL problem and $|\mathrm{Im}(H)| = |q| \approx 2m$ for the collision-resistance of the hash function, the size of the obtained signature is about twice the security level, that is one half of the size of a (EC)DSA signature with equivalent security.

4.4 Joux one-round tripartite key agreement

The Diffie-Hellman key agreement protocol enables two parties, say Alice and Bob, to establish a shared secret key over an insecure channel that may be spied on by some eavesdropper. Let $\mathbb{G} = \langle P \rangle$ be an additive group of order r on which the CDH problem is presumably hard. Alice randomly selects $a \in \mathbb{Z}_r$ and sends $[a]P$ to Bob, while Bob randomly selects $b \in \mathbb{Z}_r$ and sends $[b]P$ to Alice. The common secret is then defined as $K = [ab]P$ which can be easily computed by both Alice and Bob. An eavesdropper is then faced with computing $K = [ab]P$ from P, $[a]P$ and $[b]P$ which is precisely a CDH instance and which is hence (presumably) impossible.

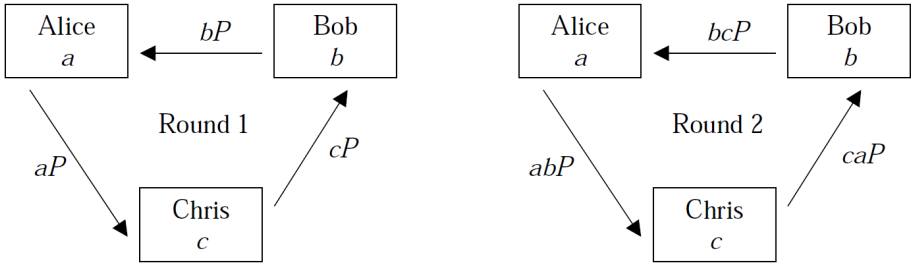

Figure 3: Two-round tripartite Diffie-Hellman protocol [Men05].

The Diffie-Hellman protocol can be extended to a tripartite protocol including a third party, say Chris. Such a two-round protocol is illustrated in Fig. 3 where Alice, Bob and Chris agreed on a shared secret $K = [abc]P$.

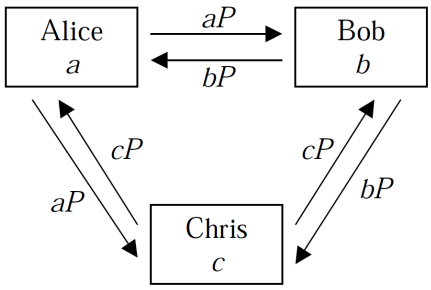

Figure 4: One-round tripartite Diffie-Hellman protocol [Men05].

In [Jou00, Jou04], Joux showed that the use of a pairing makes it possible to construct a *one-round* tripartite Diffie-Hellman protocol. This work had an important impact for cryptography as it was the first to show that pairings can be useful for the design of cryptographic protocols. The principle of Joux protocol is illustrated in Fig. 4: each party selects a random in $\mathbb{Z}_r$ and sends the corresponding

multiple of P to the other parties. Then the three parties can derive a common secret key $K = \hat{e}(P, P)^{abc}$ as:

$$K = \hat{e}([a]P, [b]P)^c = \hat{e}([a]P, [c]P)^b = \hat{e}([b]P, [c]P)^a \ .$$

An eavesdropper who wishes to recover K must then be able to compute $\hat{e}(P, P)^{abc}$ from P, $[a]P$, $[b]P$ and $[c]P$, which is precisely a BDH instance. The protocol security then holds from the hardness of BDH.

5 Pairing computation algorithms

This section describes efficient algorithms for the computation of pairings. We start by recalling Miller's algorithm on which are based all the existing algorithms to compute the Weil and the Tate pairings. We then address extension field arithmetic which is essential to pairing computation. Afterwards, we describe various optimizations of the Tate pairing and finally we describe two variants leading to faster computation in some cases.

5.1 Miller's algorithm

Let E be an elliptic curve defined over $\mathbb{F}_q$ and let r a positive integer such that $r \mid \#E(\mathbb{F}_q)$ and $\gcd(r, q) = 1$. Let k denote the embedding degree of E with respect to r. In the following, we will denote by $f_{n,P}$ any function such that:

$$\mathrm{div}(f_{n,P}) = n(P) - ([n]P) - (n-1)(\mathcal{O}) .$$

Let $P, Q \in E(\mathbb{F}_{q^k})[r]$ and $D_P, D_Q \in \mathrm{Div}(E)$ such that $D_P = (P + R) - (R)$ and $D_Q = (P + R') - (R')$ where $R, R' \in E(\mathbb{F}_{q^k})\backslash\{P, Q, \mathcal{O}\}$. Note that $D_P \sim (P) - (\mathcal{O})$ and $D_Q \sim (Q) - (\mathcal{O})$. Also note that $\mathrm{div}(f_{r,P}) = r(P) - r(\mathcal{O})$ and $\mathrm{div}(f_{r,Q}) = r(Q) - r(\mathcal{O})$. From the definition of the Weil pairing given in Section 3.2 we have:

$$w_r(P, Q) = \frac{f_{r,P}(D_Q)}{f_{r,Q}(D_P)} = \frac{f_{r,P}(Q + R')f_{r,Q}(R)}{f_{r,P}(R')f_{r,Q}(P + R)} ,$$

and from the definition of the Tate pairing given in Section 3.3 we have:

$$t_r(P, Q) = f_{r,P}(D_Q)^{(q^k-1)/r} = \left(\frac{f_{r,P}(Q + R')}{f_{r,P}(R')}\right)^{(q^k-1)/r} .$$

From these equations, it appears that computing the Weil and the Tate pairing requires an algorithm to evaluate a function $f_{r,P}$ in a point Q. Such an algorithm was proposed by Miller which is the basis of all pairing computation methods used in cryptography [Mil86, Mil04].

Let ℓ_{P_1,P_2} be a function such that $\ell_{P_1,P_2}(x, y) = 0$ is the equation of the line through P_1 and P_2 (or the tangent at $P_1 = P_2$). According to Section 2.2, this function has a zero at three points of the curve: P_1, P_2 and $-(P_1 + P_2)$. We then have:

$$\mathrm{div}(\ell_{P_1,P_2}) = (P_1) + (P_2) + (-(P_1 + P_2)) - 3(\mathcal{O}) .$$

Similarly, let v_P by any function such that $v_P(x, y) = 0$ is the equation of the vertical line through P. According to Section 2.2, this function has a zero at two points of the curve: P and $-P$. We then have:

$$\mathrm{div}(v_P) = (P) + (-P) - 2(\mathcal{O}) .$$

Miller's algorithm starts from the following observation: for every $n, m \in \mathbb{N}$ and for every $P \in E(\overline{\mathbb{F}_q})$, we have:

$$\begin{aligned}\mathrm{div}(f_{n+m,P}) &= (n+m)(P) - ([n+m]P) - (n+m-1)(\mathcal{O}) \\ &= \underbrace{(n)(P) - ([n]P) - (n-1)(\mathcal{O})}_{\mathrm{div}(f_{n,P})} \\ &\quad + \underbrace{(m)(P) - ([m]P) - (m-1)(\mathcal{O})}_{\mathrm{div}(f_{m,P})} \\ &\quad + \underbrace{([n]P) + ([m]P) - ([n+m]P) - (\mathcal{O})}_{\mathrm{div}(\ell_{[n]P,[m]P}) - \mathrm{div}(v_{[n+m]P})}\end{aligned}$$

which implies:

$$f_{n+m,P} = f_{n,P} f_{m,P} \left(\ell_{[n]P,[m]P} / v_{[n+m]P}\right) . \tag{15}$$

Also note that $\mathrm{div}(f_{0,P}) = \mathrm{div}(f_{1,P}) = 0$ which implies that $f_{0,P}$ and $f_{1,P}$ are constant and can be taken as $f_{0,P} = f_{1,P} = 1$. Equation (15) directly yields the following iterative relations:

$$\begin{cases} f_{0,P} = f_{1,P} = 1 \\ f_{i+1,P} = f_{i,P} \cdot \ell_{[i]P,P} / v_{[i+1]P} \\ f_{2i,P} = f_{i,P}^2 \cdot \ell_{[i]P,[i]P} / v_{[2i]P} \end{cases}$$

From these relations Miller derived an iterative algorithm to compute $f_{n,P}(Q)$ for any $n \in \mathbb{N}$, where every step consists in computing either $([2i]P, f_{2i,P}(Q))$ or $([2i+1]P, f_{2i+1,P}(Q))$ from $([i]P, f_{i,P}(Q))$ depending on the current bit n_i of n.

Algorithm 1 Miller's Algorithm

Input: $P \in E(\mathbb{F}_{q^k})$, $Q \in E(\mathbb{F}_{q^k})$, $n = (n_{l-1}, \ldots, n_1, n_0)_2$
Output: $f_{n,P}(Q)$ where $\mathrm{div}(f_{n,P}) = n(P) - ([n]P) - (n-1)(\mathcal{O})$

1. $T \leftarrow P$, $f \leftarrow 1$
2. **for** $i = l-1$ **downto** 0 **do**
3. $\quad f \leftarrow f^2 \cdot \ell_{T,T}(Q) / v_{[2]T}(Q)$
4. $\quad T \leftarrow [2]T$
5. $\quad$ **if** $n_i = 1$ **then**
6. $\quad\quad f \leftarrow f \cdot \ell_{T,P}(Q) / v_{T+P}(Q)$
7. $\quad\quad T \leftarrow T + P$
8. $\quad$ **end if**
9. **end for**
10. **return** f

Note that this algorithm is very close to a simple elliptic curve scalar multiplication computing $[n]P$, but with additional computation at Steps 3 and 6 (in fact Algorithm 1 computes both $[n]P$ and $f_{n,P}(Q)$ but only returns $f_{n,P}(Q)$). Moreover, as we show hereafter, part of the additional computation is already done for the computation of $[n]P$.

Detail of Steps 3 and 4. At step 3, one must compute the coefficients of $\ell_{T,T}$ and $v_{[2]T}$ in order to evaluate $\ell_{T,T}(Q)$ and $v_{[2]T}(Q)$. On the one hand, $\ell_{T,T}(x,y) = 0$ is the equation of the tangent of the curve at T. According to Section 2.2 and denoting $T = (x_T, y_T)$, we have:

$$\ell_{T,T}(x,y) = \lambda x - y + \beta \ , \quad \text{where} \quad \lambda = \frac{3x_T^2 + a}{2y_T} \quad \text{and} \quad \beta = y_T - \lambda x_T \ .$$

Still according to Section 2.2, the point $[2]T = (x', y')$ satisfies:

$$x' = \lambda^2 - 2x_T \quad \text{and} \quad y' = \lambda x' - \beta \ .$$

On the other hand $v_{[2]T}(x,y) = 0$ is the equation of the vertical line through T and v_T can be taken as $v_T(x,y) = x - x'$. The computation at Steps 3 and 4 in Algorithm 1 can then be performed as:

$\lambda \leftarrow (3x_T^2 + a)/(2y_T)$
$\beta \leftarrow y_T - \lambda x_T$
$x' \leftarrow \lambda^2 - 2x_T$
$y' \leftarrow \lambda x' - \beta$
$f \leftarrow f^2(\lambda x_Q - y_Q + \beta)/(x_Q - x')$
$T \leftarrow (x', y')$

Detail of Steps 6 and 7. At Step 6, the function $\ell_{T,P}$ is such that $\ell_{T,T}(x,y) = 0$ is the equation of the line through T and P. According to Section 2.2, we have:

$$\ell_{T,P}(x,y) = \lambda x - y + \beta \ , \quad \text{where} \quad \lambda = \frac{y_T - y_P}{x_T - x_P} \quad \text{and} \quad \beta = y_T - \lambda x_T \ .$$

Denoting, $T + P = (x', y')$, we further have:

$$x' = \lambda^2 - x_T - x_P \ \text{ and } \ y' = \lambda x' - \beta \ .$$

Finally and similarly than above, we have: $v_{P+T}(x,y) = x - x'$ and the computation at Steps 6 and 7 can be performed as:

$\lambda \leftarrow (y_T - y_P)/(x_T - x_P)$
$\beta \leftarrow y_T - \lambda x_T$
$x' \leftarrow \lambda^2 - x_T - x_P$
$y' \leftarrow \lambda x' - \beta$
$f \leftarrow f^2(\lambda x_Q - y_Q + \beta)/(x_Q - x')$
$T \leftarrow (x', y')$

Last iteration. In practice, Miller's algorithm is used to compute $f_{r,P}(Q)$ where $P, Q \in E(\mathbb{F}_{q^k})[r]$. This implies that $[r]P = \mathcal{O}$ and the last iteration of the algorithm must be slightly modified. Assume that r is odd (which is often the case in practice where r is prime), in the last iteration we have $n_0 = r_0 = 1$ and the final point which is computed (Steps 6 and 7) is $T + P = [r]P = \mathcal{O}$. Therefore, just

before this last computation, we have $T = [r-1]P = -P$, and $f = f_{r-1,P}(Q)$ where $f_{r-1,P}$ satisfies:

$$\mathrm{div}(f_{r,P}) = \mathrm{div}(f_{r-1,P}) + (P) + (-P) = \mathrm{div}(f_{r-1,P}) + \mathrm{div}(v_P)\ .$$

Steps 6 and 7 in the last iteration are then simply replaced by $f \leftarrow f \cdot v_P(Q)$ (that is $f \leftarrow f \cdot (x_Q - x_P)$).

Also, note that for Miller's algorithm to behave well, T must be different from Q and $\mathcal{O}$ at every steps of the algorithm. These requirements are always satisfied in practice where we have $Q \notin \langle P \rangle$ and $T = [i]P$ with $i < r$ at all (but the last) iterations.

Standard optmizations. Miller's algorithm can be optimized in the same ways as classical scalar multiplication algorithms for elliptic curves. One can use a NAF representation of the exponent and/or windowing techniques (see for instance [MvOV97]) or select an order r which has a low Hamming weight. Points P and Q may also be represented in projective coordinates to speed up the computation. Fast pairing computations using different coordinate systems are addressed in [IT02, CSB04, ALNR09]. The coordinate system leading the best efficiency should generally depend of the different parameters as well as on the additional optimizations which are used (see Section 5.3). A further possible optimization when the point P is fixed is to make use of precomputation. When a large amount of memory is available, one can precompute all the intermediate values of T and of the coefficients of $\ell_{T,T}$ and $v_{[2T]}$ (as we will see in Section 5.3, the latter can often be ignored). If less memory is available, one may still use the precomputation-based exponentiation techniques described [BGMW92].

In the following, we address efficient way of computing pairings based on Miller's algorithm. As this algorithm processes elements of the extension field $\mathbb{F}_{q^k}$, we first describe efficient extension field arithmetic.

5.2 Extension field arithmetic

In this section, we recall some basics about extension fields arithmetic. Every extension $\mathbb{F}_{q^k}$ of $\mathbb{F}_q$, can be represented as $\mathbb{F}_q[x]/(p(x))$ where p is a k-degree polynomial which is irreducible of $\mathbb{F}_q$. This means that every element $a \in \mathbb{F}_{q^k}$ can be seen as a polynomial $a = \sum_{i=0}^{k-1} a_i x^i$ whose coefficients a_i lie in $\mathbb{F}_q$, and hence a can be represented as a vector of k elements in $\mathbb{F}_q$: $(a_{k-1}, \ldots, a_1, a_0)$.

This representation makes clearly appear that $\mathbb{F}_{q^k}$ is a k-dimensional vector space over $\mathbb{F}_q$. In particular the sum of two elements a and b in $\mathbb{F}_{q^k}$ can be computed as $c = a + b = \sum_i c_i x^i$ where $c_i = a_i + b_i$ for every i. Also multiplying $a \in \mathbb{F}_{q^k}$ by a scalar $c \in \mathbb{F}_q$ is done by multiplying every a_i by c. On the other hand, the product of two elements a and b in $\mathbb{F}_{q^k}$ is defined as $c = ab \bmod p(x)$. Denoting $p(x) = x^k - p'(x)$ (p is k-degree and we can assume that it is unitary without loss of generality), the reduction modulo $p(x)$ consists in replacing every monomial $c_{k+i}x^{k+i}$ in the product of a and b by $c_{k+i}x^i p'(x)$ (as $x^k \equiv p'(x) \bmod p(x)$).

Example 5.1. Consider the quadratic extension $\mathbb{F}_{q^2} \simeq \mathbb{F}_q[x]/(x^2-\alpha)$ where $\alpha \in \mathbb{F}_q$ is such that $x^2-\alpha$ is irreducible. Let $a, b \in \mathbb{F}_{q^2}$ with polynomial representation $a = a_1x + a_0$ and $b = b_1x + b_0$. The product $c = ab$ satisfies:

$$\begin{aligned} c &= a_1b_1x^2 + (a_0b_1 + a_1b_0)x + a_0b_0 \bmod p(x) \\ &= (a_0b_1 + a_1b_0)x + (a_1b_1\alpha + a_0b_0) \ . \end{aligned}$$

The efficiency of such a multiplication then depends on p. For instance, if $p(x) = x^2 - \beta x - \alpha$ then the product $c = ab$ in the previous example becomes $c = (a_0b_1+a_1b_0+a_1b_1\beta)+(a_1b_1\alpha+a_0b_0)$ which requires one additional multiplication by β and one additional addition compared to the case $p(x) = x^2 - \alpha$. In general, for a k-degree extension represented as $\mathbb{F}_q[x]/(p(x))$, the more coefficients of p are 0, the more efficient the reduction is. It is clear that the degree-0 coefficient of p is non-zero otherwise $x \mid p(x)$ and p is not irreducible. Therefore, when possible, one should choose $p(x) = x^k - \alpha$. We have the following theorem from [LN97].

Theorem 5.1 (Th. 3.75 [LN97])**.** *Let $k \geq 2$ be an integer and let $\alpha \in \mathbb{F}_q^*$ with order denoted $\mathrm{ord}(\alpha)$. Then $x^k - \alpha$ is irreducible over $\mathbb{F}_q[x]$ if and only if the two following conditions are satisfied: (i) every prime factor p_i of k verifies $p_i \mid \mathrm{ord}(\alpha)$ and $p_i \nmid (q-1)/\mathrm{ord}(\alpha)$, (ii) if $4 \mid k$ then $q \equiv 1 \bmod 4$.*

Note that when α is small, a multiplication by α is far more efficient than a regular multiplication over $\mathbb{F}_q$. Therefore, in addition of choosing $p(x) = x^k - \alpha$, one shall prefer taking α as small as possible.

In what follows, we detail some efficient arithmetic over quadratic and cubic extension fields.

Quadratic extension arithmetic. Trivial implementation of the multiplication over a quadratic extension field makes use of four multiplications in the base field (see Example 5.1). It is actually possible to do it in three multiplications thanks to the Karatsuba-Ofman method which is based on the trick that once a_1b_1 and a_0b_0 have been computed, $a_0b_1 + a_1b_0$ can be computed in a single multiplication as $a_0b_1 + a_1b_0 = (a_1 + a_0)(b_1 + b_0) - a_1b_1 - a_0b_0$. The Karatsuba-Ofman method hence computes the product of two degree-2 polynomials $a_1x + a_0$ and $b_1x + b_0$ as:

$$(a_1x + a_0)(b_1x + b_0) = t_1x^2 + (t_2 - t_1 - t_0)x + t_0 \ ,$$

where $t_0 = a_0b_0$, $t_1 = a_1b_1$ and $t_2 = (a_1 + a_0)(b_1 + b_0)$.

Consider a quadratic extension $\mathbb{F}_{q^2} \simeq \mathbb{F}_q[x]/(x^2-\alpha)$. Keeping same notations as above, we get:

$$(a_1x + a_0)(b_1x + b_0) = (t_2 - t_1 - t_0)x + (t_0 + \alpha t_1) \ .$$

Assuming that α is small, the cost of the Karatsuba-Ofman method is hence of 3 multiplications over $\mathbb{F}_q$.

Using a similar trick, the square can be computed in two multiplications as:

$$(a_1x + a_0)^2 = (2t_1)x + (t_2 - (\alpha + 1)t_1) \ ,$$

where $t_1 = a_0a_1$ and $t_2 = (a_1 + a_0)(\alpha a_1 + a_0)$.

Finally the inversion in $\mathbb{F}_{q^2} \simeq \mathbb{F}_q[x]/(x^2 - \alpha)$ can be computed in two squares, two multiplications and one inversion in $\mathbb{F}_q$ as:

$$(a_1x + a_0)^{-1} = \frac{a_1^2\alpha - a_0^2}{(a_1x - a_0)} .$$

Cubic extension arithmetic. For the multiplication over cubic extension field, one can use the Toom-Cook method which computes the product of two degree-3 polynomials in five multiplications whereas the trivial method requires nine multiplications. Let $a(x) = a_2x_2 + a_1x + a_0$, let $b(x) = b_2x^2 + b_1x + b_0$, and let $c(x) = a(x)b(x) = \sum_{i=0}^{4} c_ix^i$. The Tomm-Cook algorithm interpolates the coefficients c_i by evaluating $c(x)$ at five different values: ∞, 0, 1, 2 and -1 (where $c(\infty)$ means $\lim_{x\to\infty} c(x)/x = c_4$). More precisely, we have:

$$\begin{pmatrix} a(\infty)b(\infty) \\ a(0)b(0) \\ a(1)b(1) \\ a(2)b(2) \\ a(-1)b(-1) \end{pmatrix} = \begin{pmatrix} c(\infty) \\ c(0) \\ c(1) \\ c(2) \\ c(-1) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 16 & 8 & 4 & 2 & 1 \\ -1 & 1 & -1 & 1 & -1 \end{pmatrix} * \begin{pmatrix} c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{pmatrix}$$

which gives:

$$\begin{pmatrix} c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 16 & 8 & 4 & 2 & 1 \\ -1 & 1 & -1 & 1 & -1 \end{pmatrix}^{-1} * \begin{pmatrix} a(\infty)b(\infty) \\ a(0)b(0) \\ a(1)b(1) \\ a(2)b(2) \\ a(-1)b(-1) \end{pmatrix}$$

The Toom-Cook method hence consists in computing the five products:

$$\begin{gathered} a(\infty)b(\infty) = a_2b_2,\ a(0)b(0) = a_1b_1, \\ a(1)b(1) = (a_2 + a_1 + a_0)(b_2 + b_1 + b_0), \\ a(2)b(2) = (4a_2 + 2a_1 + a_0)(4b_2 + 2b_1 + b_0), \\ \text{and } a(-1)b(-1) = (a_2 - a_1 + a_0)(b_2 - b_1 + b_0), \end{gathered}$$

from which the c_i's are then evaluated as simple linear combinations.

Considering a cubic extension $\mathbb{F}_{q^3} \simeq \mathbb{F}_q[x]/(x^3 - \alpha)$, we get:

$$ab = c_2x^2 + (\alpha c_4 + c_1)x + (\alpha c_3 + c_0) .$$

For squaring, Chung and Hasan proposed a Toom-Cook-based method which enables to square in $\mathbb{F}_{q^3}$ using four squares and one multiplication over $\mathbb{F}_q$ [CH06]. This method is therefore advantageous over the classical Toom-Cook method if squaring over $\mathbb{F}_q$ is faster than multiplying. For the inversion over $\mathbb{F}_{q^3}$ a method is given in [LH00] which requires three squares, six multiplications and one inversion over $\mathbb{F}_q$.

Lazy reduction. The above methods for fast arithmetic over extension fields can

be improved using *lazy reductions* [LH00]. If q is a prime, a multiplication over $\mathbb{F}_q$ can be computed by a multiplication over the integers followed by a reduction modulo q whose cost is usually significant compared to the multiplication cost. While adding several products or squares over $\mathbb{F}_q$ as in the above methods, one can perform the reduction modulo q only once after all non-reduced values have been added. The same principle applies with a polynomial modular reduction rather than an integer modular reduction if q is a prime power.

Pairing-friendly fields. The methods described above allow one to compute the product efficiently over any extension $\mathbb{F}_{q^k}$ where $k = 2^i 3^j$ using a *tower field representation*. Indeed, for such a value of k, $\mathbb{F}_{q^k}$ can be represented as $\mathbb{F}_{q^{k/2}}[x]/(x^2 - \alpha)$ and the product over $\mathbb{F}_{q^k}$ is computed in three multiplications over $\mathbb{F}_{q^{k/2}}$ by Karatsuba-Ofman. The field $\mathbb{F}_{q^{k/2}}$ is then represented as $\mathbb{F}_{q^{k/6}}[x]/(x^3 - \beta)$ and the product over $\mathbb{F}_{q^{k/2}}$ is computed in five multiplications over $\mathbb{F}_{q^{k/6}}$ using Toom-Cook. Iterating the same principle we get that a multiplication over $\mathbb{F}_{q^k}$ such that $k = 2^i 3^j$ can be computed with $3^i 5^j$ multiplications over $\mathbb{F}_q$. However we omit some requirement: for polynomials $(x^2 - \alpha)$ and $(x^3 - \beta)$ to exist which are irreducible over $\mathbb{F}_q$ (or over any extension of $\mathbb{F}_q$), Theorem 5.1 implies that we must have $q \equiv 1 \bmod 12$. Then α and β can be taken to be a non-square in $\mathbb{F}_q$ and a non-cube in $\mathbb{F}_q$ respectively, and to be as small as possible to get efficient modular reductions. This motivates the following definition of *pairing-friendly fields* [KM05]: an extension field $\mathbb{F}_{q^k}$ is *pairing-friendly* if $q \equiv 1 \bmod 12$ and if $k = 2^i 3^j$ for some i and j.

5.3 Tate pairing optimizations

The Tate pairing is usually considered to be more efficient than the Weil pairing. This is in part due to several optimizations which enable to substantially speed up the Tate pairing computation. In rest of this document, we therefore restrict our description to the Tate pairing and its various optimizations.

Let E be an elliptic curve defined over $\mathbb{F}_q$ and let r a positive integer such that $r \mid \#E(\mathbb{F}_q)$ and $\gcd(r, q) = 1$. Let k denote the embedding degree of E with respect to r. As we have seen in Section 5.1, the Tate pairing of two points $P, Q \in E(\mathbb{F}_{q^k})[r]$ is defined as:

$$t_r(P, Q) = \left(\frac{f_P(Q + R)}{f_P(R)} \right)^{(q^k - 1)/r}, \tag{16}$$

where f_P is any function with divisor $\mathrm{div}(f_P) = r(P) - ([r]P)$ and R is an *auxiliary point* in $E(\mathbb{F}_{q^k}) \backslash \{\mathcal{O}, P, Q\}$. The trivial way of computing the Tate pairing then consists in applying Miller's algorithm to evaluate $f_P(Q + R)/f_P(R)$ (which can be done within one single Miller loop) and in performing an exponentiation of the result to the $(q^k - 1)/r$ over $\mathbb{F}_{q^k}$.

In the following, we shall generally assume that P is a rational point (*i.e.* it lies in $E(\mathbb{F}_q)$). A direct consequence of this restriction is that only the Steps 3 and 6 in Algorithm 1 involve operations over the extension field $\mathbb{F}_{q^k}$. And more

precisely, looking at the detail of those steps (see Section 5.1) only the operations involving the coordinates of Q are operation in $\mathbb{F}_{q^k}$. The remaining operations are all performed in $\mathbb{F}_q$ and are hence substantially faster compared to $\mathbb{F}_{q^k}$ operations (recall that according to Section 5.2, even if $\mathbb{F}_{q^k}$ is a pairing-friendly field with $k = 2^i 3^j$, a multiplication over $\mathbb{F}_q$ is still $3^i 5^j$ faster than a multiplication over $\mathbb{F}_{q^k}$).

Some optimizations describe hereafter shall further require that k is even or at least greater than 1.

Auxiliary point elimination. This optimization requires P rational and $k > 1$. Then we have the following result.

Theorem 5.2 ([BKLS02])**.** *If $k > 1$ then for every $P \in E(\mathbb{F}_q)[r]$ and $Q \in E(\mathbb{F}_{q^k})$, we have:*

$$t_r(P, Q) = f_P(Q)^{(q^k-1)/r}.$$

According to the previous theorem – and assuming the embedding degree to be greater than 1 – we can eliminate the auxiliary point R is the expression of the Tate pairing (16), which makes Miller's algorithm about twice faster.

Irrelevant factors elimination. At the end of the Miller loop, once $f_P(Q)$ has been computed, it is raised to the power of $(q^k - 1)/r$. Since by definition $r \nmid q - 1$ and since $q - 1 \mid q^k - 1$, we have $q - 1 \mid (q^k - 1)/r$. This implies that the exponentiation to the $(q^k - 1)/r$ maps every element of $\mathbb{F}_q$ to 1. As a result, for every $c \in \mathbb{F}_q$, we have $(cf_P(Q))^{(q^k-1)/r} = f_P(Q)^{(q^k-1)/r}$. Therefore, every factor c appearing in the computation of $f_P(Q)$ can be discarded. In fact, this observation can be generalized to any extension of $\mathbb{F}_q$ which is a strict subfield of $\mathbb{F}_{q^k}$ (as we have $r \nmid q^m - 1$ and $q^m - 1 \mid q^k - 1$), and irrelevant factors are actually every $c \in \mathbb{F}_{q^m}$ where $m < k$.

Although the irrelevant factor elimination may not seem useful to speed up the computation described in Section 5.1, it is the basis of some of the subsequent optimizations.

Denominators elimination. The denominator elimination consists in rendering denominators in Steps 4 and 7 of Algorithm 1 irrelevant in such a way that they can be discarded from the computation. This is done by ensuring that the x-coordinate from Q lies in a strict subfield $\mathbb{F}_{q^m}$ of $\mathbb{F}_{q^k}$ which implies that $v_{[2]T}(Q)$ and $v_{T+P}(Q)$ also lie in $\mathbb{F}_{q^m}$, namely they are irrelevant. Steps 4 and 7 of Algorithm 1 can then be replaced by $f \leftarrow f^2 \cdot \ell_{T,T}(Q)$ and $f \leftarrow f \cdot \ell_{T,P}(Q)$ respectively. Ensuring that x_Q lies in $\mathbb{F}_{q^m} \subsetneq \mathbb{F}_{q^k}$ can be done in two ways depending on whether the elliptic curve E is supersingular or not.

When E is supersingular, Q is usually chosen as a point from $\mathbb{G}_2 = \langle \phi(P) \rangle$ where ϕ is a distortion map (see Section 3.4). In that case ensuring $x_Q \in \mathbb{F}_{q^d} \subsetneq \mathbb{F}_{q^k}$ can be done by choosing a distortion map which maps the x-coordinate of in $\mathbb{F}_{q^m}$ (and preferably in $\mathbb{F}_q$ which also renders the computation of $\ell_{T,T}(Q)$ and $\ell_{T,P}(Q)$ more efficient). Examples of such distortion maps are given in [BKLS02, Gal05].

When E is ordinary, Q is chosen to lie in a subgroup $\mathbb{G}_2$ of $E[r]$ which may be either the trace-zero subgroup (in the type III setting) or any other non-rational subgroup (in the type II setting). The following proposition shows that the former case enables denominator elimination when k is even.

Proposition 5.3 ([BLS03])**.** *Let k be even and let $m = k/2$. For every $Q = (x_Q, y_Q) \in E(\mathbb{F}_{q^k})[r]$, the three following properties are equivalent:*
(i) $\mathrm{Tr}(Q) = 0$,
(ii) $\Phi_q^m(Q) = -Q$,
(iii) $x_Q^{q^m} = x_Q$ *(i.e.* $x_Q \in \mathbb{F}_{q^m}$*) and* $y_Q^{q^m} = -y_Q$.

Proof. (ii) $\Rightarrow$ (i) holds by definition of the trace map. (i) $\Rightarrow$ (ii) results from the fact the trace-zero subgroup coincide with the q-eigenspace of the Frobenius which implies $\Phi_q^m(Q) = [q^m]Q$ and hence $\Phi_q^m(Q) = -Q$ as we have $q^m \equiv -1 \bmod r$. Finally, (iii) is clearly equivalent to (ii). □

Note that when $\mathbb{G}_2$ cannot be taken as the trace-zero subgroup (typically in the type II setting), one can define the pairing as $\hat{e}(P, Q) = e(P, \phi(Q))$ where $\phi = [1] - \Phi_q$. In that case, $\phi(Q) = Q - \Phi_q(Q)$ belongs to the trace-zero subgroup for every $Q \in E[r] \backslash E(\mathbb{F}_q)[r]$ (see Section 2.6).

Twisted representation. This optimization consists in taking $\mathbb{G}_2$ as the image of a subgroup of order r of a twist of E (see Section 2.5) over a strict subfield of $\mathbb{F}_{q^k}$. Assume that E has a twist of degree d with $d \mid k$ (if k is even we know that E has such a twist of degree at least 2) and denote $m = k/d$. By Proposition 2.7, E has a unique twist E' of degree d such that $r \mid \#E'(\mathbb{F}_{q^m})$ and there exists $\xi \in \sqrt[d]{\mathbb{F}_{q^m}^*} \subset \mathbb{F}_{q^k}$ such that the isomorphism $[\xi] : E' \to E : (x, y) \mapsto (x\xi^2, y\xi^3)$ maps $E'(\mathbb{F}_{q^m})[r]$ to $E(\mathbb{F}_{q^k})[r]$. The group $\mathbb{G}_2$ can then be defined as $\mathbb{G}_2 = [\xi](E'(\mathbb{F}_{q^m})[r])$. Such a choice implies a more compact representation of the points in $\mathbb{G}_2$ and a more efficient pairing computation. The efficiency of the pairing computation holds on the one hand since (assuming k is even) $x\xi^2$ lies in $\mathbb{F}_{q^{k/2}}$ which enables the denominator elimination optimization described above. By Proposition 5.3, we actually have that $\mathbb{G}_2$ is the trace-zero subgroup and the pairing is of type III (see Section 3.4). On the other hand, the point $Q \in \mathbb{G}_2 = [\xi](E'(\mathbb{F}_{q^m})[r])$ which enters in the Miller loop (see Algorithm 1) also has a compact representation which further speeds up the computation. Indeed, $\mathbb{F}_{q^k}$ is the splitting field of ξ and elements of $\mathbb{F}_{q^k}$ can be represented as polynomials of degree d over $\mathbb{F}_{q^m}[\xi]$. Then the points of $\mathbb{G}_2$ have coordinates whose representation over $\mathbb{F}_{q^m}[\xi]$ have a single non-zero monomial ($x\xi^2$ or $y\xi^3$), namely which can be represented by one single element in $\mathbb{F}_{q^m}$. These points are hence d times more compact than random points over $E(\mathbb{F}_{q^k})$, and operations involving coordinates of $Q \in \mathbb{G}_2$ in the Miller loop are faster.

Final Exponentiation. Once $f_{r,P}(Q)$ has been computed using Miller's algorithm, it must be raised to the power of $(q^k-1)/r$ to get the Tate pairing value. As $f_{r,P}(Q)$ lies in the full extension field $\mathbb{F}_{q^k}$, this final exponentiation is slow. Nevertheless, it can be significantly speeded up using standard tricks. The starting point of these

optimizations is that the Frobenius map Φ_q over $\mathbb{F}_{q^k}$ can be efficiently evaluated compared to a classical exponentiation.

Proposition 5.4 (Action of Frobenius map)**.** *Let k be a positive integer and assume that there exists $\alpha \in \mathbb{F}_q$ such that $x^k - \alpha$ is irreducible over $\mathbb{F}_q[x]$. Let $a = \sum_i a_i x^i \in \mathbb{F}_q[x]/(x^k - \alpha) \simeq \mathbb{F}_{q^k}$. The jth power Frobenius $\Phi_q^j : a \mapsto a^{q^j}$ satisfies:*

$$\Phi_q^j(a) = \sum_{i=0}^{k-1} a_i \alpha^{\lfloor iq^j/k \rfloor} x^{iq^j \bmod k} . \tag{17}$$

The straightforward evaluation of (17) is clearly faster than an exponentiation to some random exponent of size $j \log(q)$. Furthermore, it can be optimized in various ways. Assume for instance that $q \equiv 1 \bmod k$, then we have $iq^j \bmod k = i$ and $\lfloor iq^j/k \rfloor = i(q^j-1)/k$. In that case, one can compute $\beta = \alpha^{(q^j-1)/k \bmod q-1}$ using an exponentiation over $\mathbb{F}_q$ and then compute the Frobenius $\Phi_q^j(a) = \sum_{i=0}^{k-1} a_i \beta^i x^i$. As we have $\beta^{k/2} = \alpha^{(q^j-1)/2} = -1$,computing $\Phi_q^j(a)$ only requires $3k/2$ multiplications over $\mathbb{F}_q$. Another optimization consists in using some α having a small order over $\mathbb{F}_q^*$. For instance if $q \equiv 3 \bmod 4$ and if k is a power of 2, then $x^k + 1$ is irreducible over $\mathbb{F}_q[x]$ and one can take $\alpha = -1$. In that case, $\alpha^{\lfloor iq^j/k \rfloor} \in \{-1, 1\}$ and the evaluation of (17) is only a few subtractions.

Using the fact that the Frobenius computation is efficient, it is possible to speed up the final exponentiation by using multi-exponentiation techniques [Möl01]. Denoting $(q^k-1)/r = e_0 + e_1 q + e_2 q^2 + \cdots + e_{k-1} q^{k-1}$, one can precompute the f^{q^j} by $k-1$ applications of the Frobenius and then compute $f^{(q^k-1)/r} = \prod_{j=0}^{n} (f^{q^j})^{e_j}$ using a multi-exponentiation algorithm. Such an exponentiation processes about $\log q$ square-and-multiply iterations, which is around k times faster than a classical exponentiation over $\mathbb{F}_{q^k}$. However, the memory requirements of such an algorithm jeopardizes its use in constrained devices such as smart cards.

A further optimization is still possible which consists in decomposing the final exponentiation. Assume that k is even, we have $q^k - 1 = (q^{k/2} - 1)(q^{k/2} + 1)$ and by definition of k, we deduce $r \mid q^{k/2} + 1$ and therefore $f^{(q^k-1)/r} = (f^{q^{k/2}-1})^{(q^{k/2}+1)/r}$. The exponent may be further split as we have $\phi_k(q) \mid q^{k/2}+1$ where ϕ_k denotes the kth cyclotomic polynomial [3]. The final exponentiation $f \leftarrow f^{(q^k-1)/r}$ can hence be performed in three steps as:

$f \leftarrow f^{q^{k/2}-1}$
$f \leftarrow f^{(q^{k/2}+1)/\phi_k(q)}$
$f \leftarrow f^{\phi_k(q)/r}$

The first exponentiation requires a Frobenius computation and an inversion over $\mathbb{F}_{q^k}$. The inversion can be efficiently computed since $\mathbb{F}_{q^k}$ is a quadratic extension field (see Section 5.2). The second exponentiation enables to lighten the cost of the third exponentiation which is the slowest part (called the *hard exponentiation*). The second exponentiation can be processed with a few Frobenius

3. This holds since $\phi_k(q) \mid q^k - 1$ and $\phi_k(q) \nmid q^{k/2} - 1$.

computations and a few multiplications over $\mathbb{F}_{q^k}$ as we have:

$$\frac{q^{k/2}+1}{\phi_k(q)} = \begin{cases} 1 & \text{if } k \text{ is a power of 2} \\ q+1 & \text{if } k/2 \text{ is prime} \\ q^2+1 & \text{if } k=12 \\ q^3+1 & \text{if } k=18 \\ q^4+1 & \text{if } k=24 \\ \dots & \end{cases} \tag{18}$$

Note that if k is a power of 2 then we have $\phi_k(q) = q^{k/2} - 1$ and there is no second exponentiation. Otherwise, the second exponentiation enables saving $n\log(q)$ iterations in the hard exponentiation where $n \in [1,4]$ for $k \leq 24$.

Finally the third exponentiation requires a classical (multi-)exponentiation algorithm and is expected to take most of the required time for the final exponentiation. Nevertheless it can be performed with optimized squaring formulae [GPS06]. In fact, the value $f^{(q^k-1)/\phi_k(q)}$ in output of the second exponentiation belongs to the subgroup of $\mathbb{F}^*_{q^k}$ of order $\phi_k(q)$ which is called *cyclotomic subgroup of* $\mathbb{F}_{q^k}$ and denoted $G_{\phi_k(q)}$. On this subgroup, squaring can be optimized. Assume for instance that $k = 2^i 3^j$ (typically, $\mathbb{F}_{q^k}$ is a pairing-friendly field–see Section 5.2). Then the kth cyclotomic polynomial satisfies $\phi_k(q) = q^{k/3} - q^{k/6} + 1$ which implies that every $a \in G_{\phi_k(q)}$ satisfies:

$$a^{q^{k/3}} \cdot a - a^{q^{k/6}} = \sum_{i=0}^{k-1} v_i x^i = 0\,.$$

The optimization consists in rewriting the squaring formulae of $\mathbb{F}_{q^k}$ in a efficient form modulo the above equation. More precisely, we have that (i) every v_i can be expressed as a polynomial function $v_i = f_i(a_0, a_1, \dots, a_{k-1})$ where the a_i's are the coefficient of a, (ii) every v_i equals 0. Let us now denote $a^2 = \sum_{i=0}^{k-1} b_i x^i$ and $b_i = g_i(a_0, a_1, \dots, a_{k-1})$. The evaluation of each g_i in the squaring of a can be replaced by the evaluation of $g_i + \sum_j \ell_j f_j$ where the ℓ_j's are some coefficients over $\mathbb{F}_{q^k}$. This enables to derive more efficient squaring formulae over $G_{\phi_k(q)}$. Explicit formulae for squaring over cyclotomic subgroups of various extension fields are given in [SL02, GPS06, GS10].

5.4 The Eta and Ate pairings

The Eta and the Ate pairings are variants of the Tate pairing which have been introduced in [BGOS07, HSV06]. The main purpose of these alternative pairings is to speed up the computation by reducing the number of iterations in the Miller loop.

Let E be an elliptic curve defined over $\mathbb{F}_q$ and let r be a large prime such that $r \mid \#E(\mathbb{F}_q)$. Let $t = q + 1 - \#E(\mathbb{F}_q)$ be the trace of the Frobenius and let denote $T = t - 1$. Let $\mathbb{G}_1$ be the rational subgroup of the r-torsion of E *i.e.* $\mathbb{G}_1 = E[r] \cap \mathrm{Ker}(\Phi_q - [1])$ and let $\mathbb{G}_2$ to be the trace-zero subgroup *i.e.* $\mathbb{G}_2 = E[r] \cap \mathrm{Ker}(\Phi_q - [q])$.

Consider the mth power of the Tate pairing for some m:

$$t_r(P,Q)^m = f_{r,P}(Q)^{m(q^k-1)/r} = f_{mr,P}(Q)^{(q^k-1)/r}$$

where the second equality holds from $f_{mr,Q} = f_{r,Q}^m f_{m,[r]Q}$ and $[r]Q = \mathcal{O}$. As the Tate pairing is non-degenerate, its mth power is also non-degenerate as long as $m \nmid r$. Then define $m = (T^k-1)/r$ (as $T \equiv q \bmod r$ we indeed have $r \mid T^k - 1$). We get:

$$t_r(P,Q)^m = f_{T^k-1,P}(Q)^{(q^k-1)/r} = f_{T^k,P}(Q)^{(q^k-1)/r} ,$$

where the last equality holds since $\mathrm{div}(f_{T^k-1,P}) = \mathrm{div}(f_{T^k,P}) = (T^k-1)(P) - (T^k-1)(\mathcal{O})$. It can then be checked that we have:

$$f_{T^k,P} = f_{T,P}^{T^{k-1}} f_{T,[q]P}^{T^{k-2}} \cdots f_{T,[q^{k-1}]P} . \tag{19}$$

This relation is the base of the Eta and the Ate pairings.

The Eta pairing. Assume that E is *supersingular*. The Eta pairing is defined as:

$$\begin{aligned} \eta_T \ : \ \mathbb{G}_1 \times \mathbb{G}_2 &\longrightarrow \mu_r \subset \mathbb{F}_{q^k}^* \\ (P,Q) &\longmapsto f_{T,P}(Q)^{(q^k-1)/r} \end{aligned}$$

Theorem 5.5 ([BGOS07, HSV06])**.** *For every $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$, the Eta pairing satisfies:*

$$\eta_T(P,Q)^c = t_r(P,Q)^m ,$$

where $c = \sum_{i=0}^{k-1} T^{k-1-i} q^i \equiv k q^{k-1} \bmod r$. In particular, η_T is non-degenerate if and only if $m \nmid r$.

Proof. Let $\widehat{\Phi}_q$ be the dual of the Frobenius, namely $\widehat{\Phi}_q \equiv \begin{pmatrix} q & 0 \\ 0 & 1 \end{pmatrix}$ in the base (P,Q). We have:

$$f_{T,[q^i]P}(Q) = f_{T,\widehat{\Phi}_q^i(P)}(Q) = f_{T,\widehat{\Phi}_q^i(P)} \circ \widehat{\Phi}_q^i(Q) ,$$

where the first equality holds since $\widehat{\Phi}_q^i$ acts as multiplication by q^i on P, and the second equality holds since $\widehat{\Phi}_q^i$ acts trivially on Q. Then, as E is supersingular, we have $E[q^i] = \{\mathcal{O}\}$ which implies that $\widehat{\Phi}_q^i$ has a trivial kernel (since $\widehat{\Phi}_q^i = \Phi_q^{-i} \circ [q^i]$) and Proposition II.3.6 in [Sil86] yields:

$$\mathrm{div}(f_{T,\widehat{\Phi}_q^i(P)} \circ \widehat{\Phi}_q^i) = \deg(\widehat{\Phi}_q^i)\mathrm{div}(f_{T,P}) = \mathrm{div}(f_{T,P}^{q^i}) .$$

Therefore, $f_{T,[q^i]P}(Q)$ equals $f_{T,P}^{q^i}(Q)$ up to some constant in $\mathbb{F}_{q^k}$ and Theorem 5.5 then results from (19). □

The Eta pairing is evaluated by first computing $f_{T,P}(Q)$ and then raising it to the power of $(q^k-1)/r$. Compared to the Tate pairing, the computation of $f_{T,P}(Q)$ is likely to be faster than the computation of $f_{r,P}(Q)$. Indeed, if we have

$\log(r) \approx \log(q)$ then $\log(T)$ is about twice smaller than $\log(r)$ implying that the Miller loop is twice faster for the Eta pairing.

The drawback of the Eta pairing is that it is only non-degenerate for supersingular elliptic curves. If E is ordinary, then $\widehat{\Phi}_q$ does not have a trivial kernel and the above proof does not hold anymore. To overcome this issue, one can invert P and Q – *i.e.* computing $f_{T,Q}(P)$. Then the above proof holds using the Frobenius endomorphism rather that its dual. This approach yields the Ate pairing.

The Ate pairing. The Ate pairing is defined as:

$$\begin{aligned} a_T \ : \ \mathbb{G}_1 \times \mathbb{G}_2 &\longrightarrow \mu_r \subset \mathbb{F}_{q^k}^* \\ (P, Q) &\longmapsto f_{T,Q}(P)^{(q^k-1)/N} \end{aligned}$$

Theorem 5.6 ([HSV06])**.** *For every $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$, the Ate pairing satisfies:*

$$a_T(P, Q)^c = t_r(Q, P)^m \ , \tag{20}$$

where $c = \sum_{i=0}^{k-1} T^{k-1-i} q^i \equiv k q^{k-1} \bmod r$. In particular, a_T is non-degenerate if and only if $m \nmid r$.

Proof. The proof of the above theorem is the same as the proof of Theorem 5.5 but replacing the dual of the Frobenius $\widehat{\Phi}_q$ by the Frobenius Φ_q. □

Remark. While inverting the points P and Q, Theorem 5.2 does not apply anymore and for most functions $f_{r,Q}$, we have:

$$t_r(Q, P) = \left(\frac{f_{r,Q}(P + R)}{f_{r,Q}(R)} \right)^{(q^k-1)/r} \neq f_{r,Q}(P)^{(q^k-1)/r} \ .$$

Functions $f_{r,Q}$ verifying the equality are those which satisfy $f_{r,Q}(\mathcal{O}) \in \mathbb{F}_q$. Similarly, the function $f_{T,Q}$ used in the definition of the Ate pairing must satisfy $f_{T,Q}(\mathcal{O}) \in \mathbb{F}_q$ otherwise (20) does not hold. Fortunately, it is not hard to ensure $f_{T,Q}(\mathcal{O}) \in \mathbb{F}_q$ in practice. It suffices to take the y-coordinate coefficient of every line function ℓ involved in Miller's algorithm to lie in $\mathbb{F}_q$. This is the case of the description given in Section 5.1 where those coefficients are always -1.

The drawback of the Ate pairing is that computing $f_{T,Q}(P)$ is slower than computing $f_{T,P}(Q)$ as the point Q lies in the full extension field. It may even be slower than computing $f_{r,P}(Q)$ (as in the Tate pairing) depending on the values of r, T and k. One way to circumvent this drawback is by using the properties of the twist (see Section 2.5).

The twisted Ate pairing. Assume that E has a twist E' of degree d with $d \mid k$ and denote $e = k/d$. Let $\mathbb{G}_2$ be defined according to the *twisted representation* described in Section 5.3 (in particular if k is even d can be taken even and $\mathbb{G}_2$ is the trace-zero subgroup). Then, the twisted Ate pairing is defined as:

$$\begin{aligned} a_{T^e}^t \ : \ \mathbb{G}_1 \times \mathbb{G}_2 &\longrightarrow \mu_r \subset \mathbb{F}_{q^k}^* \\ (P, Q) &\longmapsto f_{T^e,P}(Q)^{(q^k-1)/r} \end{aligned}$$

Theorem 5.7 ([HSV06]). *For every $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$, the twisted Ate pairing satisfies:*

$$a^t_{T^e}(P,Q)^c = t_r(Q,P)^m \ ,$$

where $c = \sum_{i=0}^{d-1} T^{d-1-i} q^i \equiv dq^{e(d-1)} \bmod r$. In particular, $a^t_{T^e}$ is non-degenerate if and only if $m \nmid r$.

The proof of the above theorem is similar to the proof of Theorem 5.5 but the last part of the proof is based on the fact that $f_{T^e,[T^e]P}(Q) = f_{T^e,P}(Q)^{q^e}$ which holds by the properties of the twist (see [HSV06] for details).

Clearly, the twisted Ate pairing is faster than the Tate pairing if the trace is small enough to satisfy $\log(t) \leq log(r)/e$.

5.5 Generalizations and optimal pairings

Some generalizations and improvements of the Eta, Ate and twisted Ate pairings have been proposed in [MKHO07, ZZH08, Ver10].

In [MKHO07], it is argued that instead of taking T, one can take any λ such that $\lambda \equiv q \bmod r$. Then, the authors give some families of pairing-friendly curves for which $\log(q \bmod r)$ is smaller than $\log(t)$ and the choice of $\lambda = q \bmod r$ yields a faster pairing.

The authors of [ZZH08] suggest to use $\lambda_i = q^i \bmod r$ and, by a similar derivation as above, they show that:

$$(P,Q) \mapsto f_{\lambda_i,Q}(P)^{(q^k-1)/r}$$

is a non-degenerate pairing if and only if $m = (\lambda_i^{k'} - 1)/r$ is not a multiple of r where $k' = k/\gcd(i,k)$. They call this pairing the Ate$_i$ pairing (similar generalizations also hold for the Eta and the twisted Ate pairing). As shown in [ZZH08], some of the existing families of pairing-friendly curves have some λ_i which are such that $\log(\lambda_i) \leq \log(r)/\varphi(k)$ (leading to a pairing with a Miller loop which is $\varphi(k)$ times shorter than the one of the Tate pairing). In fact, since r divides $\phi_{k'}(\lambda_i)$ where $\phi_{k'}$ is the k'th cyclotomic polynomial of degree $\varphi(k')$, we have $\log(\lambda_i) \geq \log(r)/\varphi(k')$. Therefore, $\log(r)/\varphi(k)$ Miller iterations is actually the best that can be obtained following the above approach. This motivates the definition of *optimal pairings* in [Ver10] as pairings which can be evaluated in $\log(r)/\varphi(k) + \varepsilon(k)$ Miller iterations (with $\varepsilon(k) \leq log(k)$).

Finally [Ver10] further generalizes the Ate pairing approach by considering mth powers of the Tate pairing for some m such that mr have base-q expansion with small coefficients.

Theorem 5.8 ([Ver10]). *Let $\lambda = mr$ with $r \nmid m$ and write $\lambda = \sum_{i=0}^{l} c_i q^i$ then:*

$$\begin{aligned} a_{(c_i)_i} \ : \ \mathbb{G}_1 \times \mathbb{G}_2 &\longrightarrow \mu_r \subset \mathbb{F}_{q^k}^* \\ (P,Q) &\longmapsto \left(\prod_{i=0}^{l} f_{c_i,Q}(P)^{q^i} \cdot \prod_{i=0}^{l-1} \ell_i(P)/v_i(P) \right)^{(q^k-1)/r} \end{aligned}$$

where $\ell_i = \ell_{[s_{i+1}]Q,[c_i q^i]Q}$, $v_i = v_{[s_i]Q}$ *and* $s_i = \sum_{j=i}^{l} c_j q^j$, *defines a pairing which is non-degenerate if and only if:*

$$mkq^{k-1} \not\equiv ((q^k - 1)/r) \sum_{i=0}^{l} i c_i q^{i-1} \bmod r \ .$$

Note that if k is even, then the denominator elimination applies and the v_i's can be ignored. Also, in the computation of $[c_i q^i]Q$ (which is required to evaluate $\ell_i(P)$), one should replace all the multiplications by q^i by Frobenius actions.

The approach proposed in [Ver10] then consists in searching a λ having a q-base expansion with small coefficients c_i's to render the computation of the above pairing efficient. An algorithm is proposed to perform such a search and strong evidence is given than the lower bound for the number of Miller iterations in the obtained pairings is around $\log(r)/\varphi(k)$. This confirms the optimality property of pairings reaching this value. Finally [Ver10] shows that several existing families of pairing-friendly elliptic curves have such optimal pairings and give their formulae.

6 Summary and recommendations

A pairing is a function e which maps a pair of points of an elliptic curve E defined over some finite field $\mathbb{F}_q$ to the multiplicative group of a finite extension $\mathbb{F}_{q^k}$. Moreover, a pairing is bilinear – $e([a]P, [b]Q) = e(P, Q)^{ab}$ for all points P and Q on the curve and for all integers a and b – and it is non-degenerate – $e(P, Q) \neq 1$ for some P and Q. Famous examples of pairings include the Weil and the Tate pairings which are developed in Section 3. Pairings used in cryptography are usually restricted to a pair of subgroups $\mathbb{G}_1$ and $\mathbb{G}_2$ of the points of E which are chosen to have a large prime order r dividing $\#E(\mathbb{F}_q)$, so that the discrete logarithm is hard to compute on these groups. The embedding degree k of E with respect to r is the smallest integer such that the set of all the points of order r in E is included in $E(\mathbb{F}_{q^k})$ (which is also the smallest integer such that $r \mid q^k - 1$). This set, denoted $E[r]$ and called the r-torsion of E, contains $r^2 + 1$ distinct subgroups of order r, among which, one and only one is included in $E(\mathbb{F}_q)$. One usually takes $\mathbb{G}_1$ to be the latter subgroup while $\mathbb{G}_2$ is taken as a distinct subgroup of $E[r]$. Then the Weil and the Tate pairings defined non-degenerate bilinear maps from $\mathbb{G}_1 \times \mathbb{G}_2$ into $\mathbb{G}_T = \mu_r$, the subgroup of $\mathbb{F}^*_{q^k}$ containing all the rth roots of unity of $\mathbb{F}_q$. It is interesting to note that, by primality of r, there essentially exists one single pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. More precisely, every pairing $e' : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a power of e and there exist $r - 1$ different non-degenerate powers (including the restrictions of the Weil and the Tate pairings to $\mathbb{G}_1 \times \mathbb{G}_2$).

Pairings have originally been used to attack elliptic curve cryptosystems as they allow to transport the discrete logarithm problem from $E[r]$ to $\mu_r \subset \mathbb{F}^*_{q^k}$ where it may be more easy to solve (depending on the value of k). Nevertheless, it soon appeared that pairings have much greater potential for constructive applications in cryptography. As explained in Section 4, new intractable problems based on pairings were revealed making possible to construct new cryptographic primitives such as one-round tripartite key-agreement, identity-based encryption and short digital signatures. However, an important issue arose with the emergence of pairing-based cryptography, which regards the efficiency of the computation of pairings. Straightforward implementations of the Weil or the Tate pairing are rather inefficient which was considered at the beginning as a major drawback of pairing-based protocols.

Motivated by this issue, a consistent research work has been done since then to optimize the computation of pairings. Several improvements have been published making the computation of pairings much more efficient. As explained in Section 5, promising algorithms for computing pairings proposed so far are all based on Miller's algorithm which is a kind of binary exponentiation. In particular, this algorithm can be seen as a scalar multiplication of the point P by the order r with some additional computation at every loop iteration. This additional computation involves the point Q as well as the intermediate results of the scalar multiplication of P to update a value which equals $f_P(Q)$ at the end of the loop (where $f_P : E(\mathbb{F}_{q^k}) \to \mathbb{F}_{q^k}$ is a function parametrized by P). The pairing value is either a power of $f_P(Q)$ or a product of several such functions (in which case several Miller loops are required). A first way to optimize the computation then consists in using standard optimizations of exponentiation algorithms (exponent recoding,

windowing, ...) or in choosing the order r to have a small Hamming weigh. Also, several coordinate systems can be used for points P and Q. More generally, the pairing arithmetic is composed of the classical arithmetic over $E(\mathbb{F}_q)$ (including operations over $\mathbb{F}_q$), as well as operations over $\mathbb{F}_{q^k}$ which are necessary to the update of $f_P(Q)$ at every loop iteration. Elements in $\mathbb{F}_{q^k}$ are represented as vectors with k coordinates over $\mathbb{F}_q$ and operations over $\mathbb{F}_{q^k}$ are naturally more costly than operations over $\mathbb{F}_q$. For instance a multiplication over $\mathbb{F}_{q^k}$ usually requires from $k^{1.5}$ to k^2 multiplications over $\mathbb{F}_q$, plus several additions. To obtain a fast pairing computation it is therefore important to choose an extension field $\mathbb{F}_{q^k}$ with efficient arithmetic (*e.g.* a pairing-friendly field). In such a context, the Tate pairing appeared to be more efficient to compute than the Weil pairing. The former, which is defined as $(P, Q) \mapsto f_P(Q)^{(q^k-1)/r}$, was further optimized in various ways. It was shown that part of the computation over $\mathbb{F}_{q^k}$ in Miller's algorithm (*the denominators*) can be removed when k is even. Also, Q has a *twisted representation* with coordinates over $\mathbb{F}_{q^{k/d}}$ when E admits a twist of degree d, for some $d \leq 6$. Using such a representation speeds up the computation and decreases the memory consumption. On the other hand, the final exponentiation can take advantage of fast Frobenius computation and efficient cyclotomic subgroup arithmetic to achieve good timings. Finally, some variants of the Tate pairing have been introduced with a lower number of Miller loop iterations. It was then argued that *optimal pairings* exist which can be computed within $\log(r)/\varphi(k)$ Miller loop iterations (compared to $\log(r)$ for the Tate pairing).

All these optimizations can render a pairing computation fairly efficient. This is even true in a constrained environment such as a smart-card, provided that it at least includes a hardware co-processor for the multiplication over $\mathbb{F}_q$. The required security level yields a minimum size for q and a minimum value for k. The actual parameters can then be chosen in order to make the various optimizations possible. In general, one shall prefer a pair (q, k) for which $\mathbb{F}_{q^k}$ has efficient arithmetic, an even k for so-called denominator elimination in the Miller loop, a curve E which has a 6th degree twist (and a k multiple of 6) for efficient twisted representation, and a curve E which has optimal pairings.

References

[ALNR09] Christophe Arene, Tanja Lange, Michael Naehrig, and Christophe Ritzenthaler. Faster Computation of the Tate Pairing. Cryptology ePrint Archive, Report 2009/155, 2009. `http://eprint.iacr.org/`. To appear in Journal of Number Theory.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-Based Encryption from the Weil Pairing. In J. Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.

[BF03] Dan Boneh and Matthew K. Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.

[BGMW92] Ernest F. Brickell, Daniel M. Gordon, Kevin S. McCurley, and David Bruce Wilson. Fast Exponentiation with Precomputation (Extended Abstract). In Rainer A. Rueppel, editor, *Advances in Cryptology – EUROCRYPT '92*, volume 658 of *Lecture Notes in Computer Science*, pages 200–207. Springer, 1992.

[BGOS07] Paulo S. L. M. Barreto, Steven D. Galbraith, Colm O'Eigeartaigh, and Michael Scott. Efficient pairing computation on supersingular Abelian varieties. *Desings, Codes and Cryptography*, 42(3):239–271, 2007.

[BIPV10] Anja Becker, Sorina Ionica, Jérôme Plût, and Karine Villegas. Review of cryptographic protocols based on elliptic curves. Technical report, ANR Project ECLIPSES, 2010. Deliverable 1.1.

[BK98] R. Balasubramanian and Neal Koblitz. The Improbability That an Elliptic Curve Has Subexponential Discrete Log Problem under the Menezes–Okamoto–Vanstone Algorithm. *Journal of Cryptology*, 11(2):141–145, 1998.

[BKLS02] Paulo S. L. M. Barreto, Hae Yong Kim, Ben Lynn, and Michael Scott. Efficient Algorithms for Pairing-Based Cryptosystems. In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 354–368. Springer, 2002.

[BL96] Dan Boneh and Richard J. Lipton. Algorithms for Black-Box Fields and their Application to Cryptography (Extended Abstract). In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 283–297. Springer, 1996.

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. Short Signatures from the Weil Pairing. In E. Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 514–532. Springer, 2001.

[BLS03] Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. On the Selection of Pairing-Friendly Groups. In Mitsuru Matsui and Robert J.

Zuccherato, editors, *Selected Areas in Cryptography – SAC 2003*, volume 3006 of *Lecture Notes in Computer Science*, pages 17–25. Springer, 2003.

[BLS04] Dan Boneh, Ben Lynn, and Hovav Shacham. Short Signatures from the Weil Pairing. *Journal of Cryptology*, 17(4):297–319, 2004.

[CH06] Jaewook Chung and M. Anwar Hasan. Asymmetric squaring formulae. Technical report, University of Waterloo, 2006.

[CSB04] Sanjit Chatterjee, Palash Sarkar, and Rana Barua. Efficient Computation of Tate Pairing in Projective Coordinate over General Characteristic Fields. In Choonsik Park and Seongtaek Chee, editors, *Information Security and Cryptology – ICISC 2004*, volume 3506 of *Lecture Notes in Computer Science*, pages 168–181. Springer, 2004.

[dB88] Bert den Boer. Diffie-Hillman is as Strong as Discrete Log for Certain Primes. pages 530–539, 1988.

[FR94] Gerhard Frey and Hans-Georg Rück. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Math. Comput.*, 62(206):865–874, 1994.

[FST10] David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. *Journal of Cryptology*, 23(2):224–280, 2010.

[Gal05] Steven D. Galbraith. *Pairings*, chapter IX, pages 183–213. London Mathematical Society Lecture Note Series. Cambridge University Press, 2005.

[GPS06] S.D. Galbraith, K.G. Paterson, and N.P. Smart. Pairings for cryptographers. Cryptology ePrint Archive, Report 2006/165, 2006. `http://eprint.iacr.org/`.

[GR04] Steven D. Galbraith and Victor Rotger. Easy Decision-Diffie-Hellman Groups. *LMS Journal of Computation and Mathematics*, 7, 2004.

[GS10] Robert Granger and Michael Scott. Faster Squaring in the Cyclotomic Subgroup of Sixth Degree Extensions. In Phong Q. Nguyen and David Pointcheval, editors, *Public Key Cryptography – PKC 2010*, volume 6056 of *Lecture Notes in Computer Science*, pages 209–223. Springer, 2010.

[HSV06] Florian Hess, Nigel P. Smart, and Frederik Vercauteren. The Eta Pairing Revisited. *IEEE Transactions on Information Theory*, 52(10):4595–4602, 2006.

[IT02] Tetsuya Izu and Tsuyoshi Takagi. Efficient Computations of the Tate Pairingfor the Large MOV Degrees. In Pil Joong Lee and Chae Hoon Lim, editors, *Information Security and Cryptology – ICISC 2002*, volume 2587 of *Lecture Notes in Computer Science*, pages 283–297. Springer, 2002.

[Jou00] Antoine Joux. A One Round Protocol for Tripartite Diffie-Hellman. In Wieb Bosma, editor, *Algorithmic Number Theory – ANTS-IV*, volume 1838 of *Lecture Notes in Computer Science*, pages 385–394. Springer, 2000.

[Jou04] Antoine Joux. A One Round Protocol for Tripartite Diffie-Hellman. *Journal of Cryptology*, 17(4):263–276, 2004.

[Joy95] Marc Joye. Introduction à la théorie des courbes elliptiques. Master thesis, UCL, 1995. Available at `http://joye.site88.net/publications.html`.

[KM05] Neal Koblitz and Alfred Menezes. Pairing-Based Cryptography at High Security Levels. volume 3796 of *Lecture Notes in Computer Science*, pages 13–36. Springer, 2005.

[LH00] Chae Hoon Lim and Hyo Sun Hwang. Fast Implementation of Elliptic Curve Arithmetic in GF(p^n). In Hideki Imai and Yuliang Zheng, editors, *Public Key Cryptography – PKC 2000*, volume 1751 of *Lecture Notes in Computer Science*, pages 405–421. Springer, 2000.

[LN97] Rudolf Lidl and Harald Niederreiter. *Finite fields*, volume 20 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, second edition, 1997.

[Men05] Alfred Menezes. An introduction to pairing-based cryptography., 2005.

[Mil86] Victor S. Miller. Short Programs for functions on Curves. Technical report, IBM Watson Research Center, 1986.

[Mil04] Victor S. Miller. The Weil Pairing, and Its Efficient Calculation. *Journal of Cryptology*, 17(4):235–261, 2004.

[MKHO07] Seiichi Matsuda, Naoki Kanayama, Florian Hess, and Eiji Okamoto. Optimised Versions of the Ate and Twisted Ate Pairings. In Steven D. Galbraith, editor, *Cryptography and Coding, IMA International Conference*, volume 4887 of *Lecture Notes in Computer Science*, pages 302–312. Springer, 2007.

[Möl01] Bodo Möller. Algorithms for Multi-exponentiation. In S. Vaudenay and A.M. Youssef, editors, *Selected Areas in Cryptography – SAC 2001*, volume 2259, pages 165–180, 2001.

[MOV91] Alfred Menezes, Tatsuaki Okamoto, and Scott A. Vanstone. Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field. In *ACM Symposium on Theory of Computing – STOC '91*, pages 80–89. ACM, 1991.

[MOV93] Alfred Menezes, Tatsuaki Okamoto, and Scott A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39(5):1639–1646, 1993.

[MvOV97] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

[MW99] Ueli Maurer and Stefan Wolf. The Relationship Between Breaking the Diffie-Hellman Protocol and Computing Discrete Logarithms. *SIAM Journal on Computing*, 28(5):1689–1721, 1999.

[Sha84] Adi Shamir. Identity-Based Cryptosystems and Signature Schemes. In G.R. Blakley and D. Chaum, editors, *Advances in Cryptology – CRYPTO '84*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984.

[Sil86] J.H. Silverman. *The Arithmetic of Elliptic Curves*, volume 106 of *Graduate Texts in Mathematics*. Springer-Verlag, 1986.

[SL02] Martijn Stam and Arjen K. Lenstra. Efficient Subgroup Exponentiation in Quadratic and Sixth Degree Extensions. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *CHES*, volume 2523 of *Lecture Notes in Computer Science*, pages 318–332. Springer, 2002.

[Ver01] Eric R. Verheul. Evidence that XTR Is More Secure than Supersingular Elliptic Curve Cryptosystems. In Birgit Pfitzmann, editor, *Advances in Cryptology - EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 195–210. Springer, 2001.

[Ver04] Eric R. Verheul. Evidence that XTR Is More Secure than Supersingular Elliptic Curve Cryptosystems. *Journal of Cryptology*, 17(4):277–296, 2004.

[Ver10] Frederik Vercauteren. Optimal Pairings. *IEEE Transactions on Information Theory*, 56(1):455–461, 2010.

[Wik] Wikipedia. Elliptic curve. `http://en.wikipedia.org/wiki/Elliptic_curve`.

[ZZH08] Chang-An Zhao, Fangguo Zhang, and Jiwu Huang. A note on the Ate pairing. *Journal Information Security*, 7(6):379–382, 2008.