Adequate Elliptic Curves for Computing the Product of n Pairings

Loubna Ghammam[1,2](✉) and Emmanuel Fouotsa[3,4]

[1] IRMAR, UMR CNRS 6625, Université Rennes 1, Campus de Beaulieu, 35042 Rennes Cedex, France
ghammam.loubna@yahoo.fr

[2] Laboratoire d'électronique et de microélectronique, FSM Monastir Université de Monastir, Monastir, Tunisia

[3] LMNO, UMR CNRS 5139 Université de Caen, Campus 2, 14032 Caen Cedex, France
emmanuelfouotsa@yahoo.fr

[4] Higher Teacher Training College, University of Bamenda, P.O. Box 39, Bambili, Cameroon

Abstract. Many pairing-based protocols require the computation of the product and/or of a quotient of n pairings where $n > 1$ is a natural integer. Zhang et *al.* [1] recently showed that the Kachisa-Schafer and Scott family of elliptic curves with embedding degree 16 denoted KSS16 at the 192-bit security level is suitable for such protocols comparatively to the Baretto-Lynn and Scott family of elliptic curves of embedding degree 12 (BLS12). In this work, we provide important corrections and improvements to their work based on the computation of the optimal Ate pairing. We focus on the computation of the final exponentiation which represent an important part of the overall computation of this pairing. Our results improve by 864 multiplications in $\mathbb{F}_p$ the computations of Zhang et *al.* [1]. We prove that for computing the product or the quotient of 2 pairings, BLS12 curves are the best solution. In other cases, especially when $n > 2$ as mentioned in [1], KSS16 curves are recommended for computing product of n pairings. Furthermore, we prove that the curve presented by Zhang et *al.* [1] is not resistant against small subgroup attacks. We provide an example of KSS16 curve protected against such attacks.

Keywords: BN curves · KSS16 curves · BLS curves · Optimal Ate pairing · Product of n pairings · Subgroup attacks

1 Introduction

Pairing-based cryptography is another way of building cryptographic protocols. Thanks to the various and steady improvements for the computation of pairings

This work was supported in part by French ANR projects PEACE and ANR-12-INSE-0014 SIMPATIC, LIRIMA MACISA project and centre Henri Lebesgue, The Simons Foundations through Pole of Research in Mathematics with applications to Information Security, Subsaharan Africa.

S. Duquesne and S. Petkova-Nikova (Eds.): WAIFI 2016, LNCS 10064, pp. 36–53, 2016.
DOI: 10.1007/978-3-319-55227-9_3

on elliptic curves together with their implementation, several protocols have been published [2–6]. The BN [7] family of elliptic curves are the most suitable for implementing pairing-based cryptography at the 128-bit security level. At the high security level, the BLS12 [8] curves are recommended for computing the optimal Ate pairing according to the results presented in [9,10].

Many pairing-based protocols require the computation of products or quotients of pairings. Some of them require the computation of two pairings [11], others require three pairings [12] and even more than three pairings as in [13,14]. The few works that studied an efficient computation of products of pairings are those of Granger and Smart [1,15]. In particular, Zhang et *al.* [1] have recently shown that the KSS16 [16] elliptic curves are more suitable when computing products or quotients of optimal Ate pairings at the 192-bit security level. In their work they gave explicit formulas and cost evaluation for the Miller loop and developed interesting ways of computing the hard part of the final exponentiation. Unfortunately their results contain several forgotten operations costing 1332 multiplications in the base field $\mathbb{F}_p$. In this work we study the computation of the optimal Ate pairing on KSS16 curves. We present also a new multiple of the hard part of the final exponentiation of the optimal Ate pairing. This new multiple enabled us to improve the cost of the computation of the hard part of the final exponentiation with respect to the work of Zhang et *al.* [1]. We also compare the efficiency of KSS16 curves when computing product of pairings with respect to other common curves at the same security level. We also analyzed the resistance of the KSS16 curves to the small subgroup attack following the approach described in [17]. More precisely, the contribution of this work is as follows:

1. We first pointed out ignored operations in the computation of the optimal Ate pairing (final exponentiation) on KSS16 curves by Zhang et *al.* [1] and give detailed cost of operations with a magma code to verify the formulas [18]. Despite the improvement we obtained for the computation of the final exponentiation in this case and based on the fastest known result to date to our knowledge, we show that BLS12 curves are suitable for the computation of products of two pairings at the high security level and not KSS16 curves as recommended in [1]. We also proved that for computing n pairings where $n > 2$ then KSS16 curves are the best solution.
2. In [17], Barreto et *al.* recently studied the resistance of BN, BLS and KSS18 curves to small subgroup attacks. We extend the same analysis to KSS16 curves. In particular we show that the parameters used in [1] do not ensure protection of these curves to such attacks and we provide an example of KSS16 curve resistant to this attack.

The rest of this work is organized as follows: Sect. 2 recalls results from [1] on optimal Ate pairing on KSS16 curves. We point out the forgotten operations and bring corrections and improvements in the computation of the final exponentiation. In Sect. 3, we present our new multiple of the hard part of the final exponentiation d'. We prove that by using the new vector we saved 864M with respect to the corrected work of Zhang et *al.* in the computation of the optimal

Ate pairing over KSS16 curves. Section 4 defines products of pairings and their efficient computation. Detailed costs of the calculation and comparison are then done with commonly pairing-friendly curves at the high security level. The Sect. 5 concerns the resistance of the KSS16 curves against small subgroup attacks. We show that the curve used in [1] is not protected against small subgroup attack and provide an adequate example. We conclude our work in Sect. 6.

Notations: In this paper we denote by:

- M_k a multiplication in $\mathbb{F}_{p^k}$.
- S_k a squaring in $\mathbb{F}_{p^k}$.
- F_k a Frobenius map in $\mathbb{F}_{p^k}$.
- I_k an inversion in $\mathbb{F}_{p^k}$.
- S_c a cyclotomic squaring in $\mathbb{F}_{p^{16}}$.
- C_c a cyclotomic cube in $\mathbb{F}_{p^{16}}$.

A multiplication, a square and an inversion in $\mathbb{F}_p$ are denoted respectively by M, S and I.

2 Pairings at High Security Level

The 192-bit security level is one of the highest security level recommended when implementing cryptographic protocols based on pairings. Aranha et *al.* [9] recommended the implementation of optimal Ate pairing at this security level over BLS12 curves. Their results on BLS12 curves have been improved by Ghammam and Fouotsa in [10] and still confirm that BLS12 curves are a better solution for implementation at the 192-bit security level. Recently, Zhang et *al.* [1] considered the computation of the optimal Ate pairing over KSS16 curves at the same security level. They proved in particular that this family of curves is suitable for computing products or quotients of pairings generally involved in many pairing-based protocols. In this section we review their computation of the optimal Ate pairing and in particular we bring corrections to shortcomings in their work and give improvements in the computation of the hard part of the final exponentiation. The previous data on costs of computing optimal Ate pairing from the literature at the 192-security level are given in Table 1.

Table 1. Latest best costs of optimal Ate pairing at the 192-bit security level.

Elliptic curves	Size of p (bit)	Complexity of Miller loop	Complexity of the final exponentiation
BLS12 Curves [10]	640	10785M	8116M+6I
BLS24 Curves [10]	480	14574M	23864M+10I
BN Curves [9]	640	16553M	7218M+4I
KSS18 Curves [9]	480	13168M	23821M+8I

Remark 1. *Recently, Kim presented in [19] improvements in discrete logarithm computation in finite fields of the form $\mathbb{F}_{p^{12}}$. Then Jeong and Kim generalized it in [20]. They proved the same result for any composite extension degree n when the prime p is of a special form which is the case of BN, BLS and KSS curves which we studied in this paper. Therefore, these curves no longer provide a 192-bit security level. However, they still present a high security level since it is more than the 128-bit security level.*

2.1 The KSS16 Family of Elliptic Curves and Optimal Ate Pairing

Kachisa et al. proposed in [16] a family of pairing-friendly elliptic curves of embedding degree $k \in \{16, 18, 32, 36, 40\}$. The main idea of their construction of these families of curves is to use the minimal polynomial of the elements of the cyclotomic field rather than the cyclotomic polynomial $\phi_k(x)$ to define the cyclotomic field.

The family of curve with $k = 16$ which is called KSS16 curves is parameterised as follows:

$$\begin{cases} t = 1/35\left(2u^5 + 41u + 35\right) \\ r = u^8 + 48u^4 + 625 \\ p = \dfrac{1}{980}(u^{10} + 2u^9 + 5u^8 + 48u^6 + 152u^5 + 240u^4 + 625u^2 + \\ \quad 2398u + 3125) \end{cases} \tag{1}$$

and the equation of the elliptic curve defined over $\mathbb{F}_p$ is of the form

$$y^2 = x^3 + ax$$

where t is the trace of the Frobenius endomorphism on E, p is the field size and r presents the order the pairing-friendly subgroup. Let $G_1 = E(\mathbb{F}_p)[r]$ be the r-torsion subgroup of $E(\mathbb{F}_p)$ and $G_2 = E'(\mathbb{F}_{p^4})[r] \cap \operatorname{Ker}(\pi_p - [p])$ where E' is the quartic twist of E. The subgroup of $\mathbb{F}^{\star}_{p^{16}}$ consisting of r-th roots of unity is denoted by $G_3 = \mu_r$. Consider the function $f_{u,Q}$ with divisor $\operatorname{Div}(f_{u,Q}) = u(Q) - ([u]Q) - (u-1)(\mathcal{O})$ and $\ell_{R,S}$ the straight line passing through the points R and S of the elliptic curve.

Proposition 2. [1] *The optimal Ate pairing on the KSS16 curves is the bilinear and non degenerated map:*

$$\begin{aligned} e_{opt} : G_1 \times G_2 &\rightarrow G_3 \\ (P, Q) &\longmapsto \left((f_{u,Q}(P) l_{[u]Q,[p]Q}(P))^{p^3} l_{Q,Q}(P)\right)^{\frac{p^{16}-1}{r}} \end{aligned}$$

The parameter u proposed by Zhang et *al.* [1] is

$$u = 2^{49} + 2^{26} + 2^{15} - 2^7 - 1$$

which is a 49-bit integer of Hamming weight equal to 5 so that r has a prime factor of 377 bits and p is a prime integer of 481 bits. The computation of pairing involves two main steps: the Miller loop and the final exponentiation.

2.2 The Miller Loop

In our case, to compute the optimal Ate pairing in Proposition 2, the Miller loop consists of the computation of $(f_{u,Q}(P) \cdot l_{[u]Q,[p]Q}(P))^{p^3} \cdot l_{Q,Q}(P)$. Let $u = u_n 2^n + \cdots + u_1 2 + u_0$ with $u_i \in \{-1, 0, 1\}$. The computation of the function $f_{u,Q}(P)$ is done thanks to the algorithm in Table 2 known as the Miller algorithm [21]. The Miller loop consists of computing $f_{u,Q}(P)$, $l_{[u]Q,[p]Q}(P)$, $l_{Q,Q}(P)$ and two sparse multiplications in $\mathbb{F}_{p^{16}}$ to multiply terms together and one p^3-Frobenius.

Table 2. Miller algorithm.

	Miller algorithm: **Input**: $u = (u_n, u_{n-1}, \ldots, u_0)$,$P$,$Q$, **Output**:$(f_{u,Q}(P) \cdot l_{[u]Q,[p]Q}(P))^{p^3} \cdot l_{Q,Q}(P)$		
1:	Set $f_1 \leftarrow 1$ and $R \leftarrow Q$		
2:	**For** $i = n-1$ **down to** 0 **do**		
3:	$f_1 \leftarrow f_1^2 \cdot \ell_{R,R}(P)$,	$R \leftarrow 2R$	Doubling step
5:	**if** $u_i = 1$ **then**		
6:	$f_1 \leftarrow f_1 \cdot \ell_{R,Q}(P)$	$R \leftarrow R + Q$, **end if**	Addition step
7:	**if** $u_i = -1$ **then**		
8:	$f_1 \leftarrow f_1 \cdot \ell_{R,-Q}(P)$	$R \leftarrow R - Q$, **end if**	Addition step
9:	**end For**		
10:	**return** $f_1 = f_{u,Q}(P)$		

The computation of $f_{u,Q}(P)$ costs 49 doubling steps with associated line evaluation, 4 addition steps with line evaluations, 48 squarings in $\mathbb{F}_{p^{16}}$ and 52 sparse multiplications in $\mathbb{F}_{p^{16}}$. We then need an extra 2p-Frobenius maps for computing $[p]Q$ and $[u]Q$ is obtained through the computation of $f_{u,Q}(P)$. Thus we have to perform 8 multiplications in $\mathbb{F}_p$, a multiplication in $\mathbb{F}_{p^4}$ and one squaring in $\mathbb{F}_{p^4}$ plus 2p-Frobenius to compute $l_{[u]Q,[p]Q}(P)$. We need also 8 multiplications in $\mathbb{F}_p$, 4 multiplications in $\mathbb{F}_{p^4}$, and one squaring in $\mathbb{F}_{p^4}$ to compute $l_{Q,Q}(P)$ (see [1] for formulas and complete details on the costs).

Therefore, the overall cost of the computation of the Miller loop, as mentioned in [1], is 49 doubling steps with associated line evaluations, 4 addition steps with line evaluations, 48 squarings in $\mathbb{F}_{p^{16}}$, 54 sparse multiplications in $\mathbb{F}_{p^{16}}$, 2p, p^3 Frobenius maps in $\mathbb{F}_{p^{16}}$, 16 multiplications in $\mathbb{F}_p$, 5 multiplications in $\mathbb{F}_{p^4}$ and one squaring in $\mathbb{F}_{p^4}$. From Table 4 of [1], the Miller loop of the optimal Ate pairing on KSS16 curve costs about 10208 multiplications in $\mathbb{F}_p$.

2.3 The Final Exponentiation

The second step in computing the optimal Ate pairing is the final exponentiation which consists of raising the result f_1 of the Miller loop to the power $\frac{p^{16}-1}{r}$. Thanks to the cyclotomic polynomial, this expression is simplified and presented as follows:

$$f_1^{\frac{p^{16}-1}{r}} = (f_1^{p^8-1})^{\frac{p^8+1}{r}}.$$

First we have to compute $f = f_1^{p^8-1}$ which is called the simple part of the final exponentiation. This costs one p^8–Frobenius, an inversion and a multiplication in $\mathbb{F}_{p^{16}}$. Raising f to the power $\frac{p^8+1}{r}$ is called the hard part of the final exponentiation. In [1], Zhang et *al.* considered a multiple of the second part of the final exponentiation. So instead of computing f^d they computed $f^{857500d}$ where $d = \frac{p^8+1}{r}$. This choice enables them to only have integer coefficients in the representation of $d_1 = 857500d$ in base p which is a simple way for computing this hard part of the final exponentiation.

$$\frac{p^8+1}{r} = \sum_{i=0}^{\phi(16)-1} c_i p^i = c_0 + c_1 p + c_2 p^2 + \cdots + c_7 p^7$$

Where:

$$\begin{cases} c_0 = -11u^9 - 22u^8 - 55u^7 - 278u^5 - 1172u^4 - 1390u^3 + 1372 \\ c_1 = 15u^8 + 30u^7 + 75u^6 + 220u^4 + 1280u^3 + 1100u^2 \\ c_2 = 25u^7 + 50u^6 + 125u^5 + 950u^3 + 3300u^2 + 4750u \\ c_3 = -125u^6 - 250u^5 - 625u^4 - 3000u^2 - 13000u - 15000 \\ c_4 = -2u^9 - 4u^8 - 10u^7 + 29u^5 - 54u^4 + 154u^3 + 4704 \\ c_5 = -20u^8 - 40u^7 - 100u^6 - 585u^4 - 2290u^3 - 2925u^2 \\ c_6 = 50u^7 + 100u^6 + 250u^5 + 1025u^3 + 4850u^2 + 5125u \\ c_7 = 875u^2 + 1750u + 4375 \end{cases} \quad (2)$$

Then Zhang et *al.* presented a very nice decomposition of c_i where $i \in \{0, 1, 2, 3, 4, 5, 6, 7\}$. This representation enabled them to quickly compute the hard part of the final exponentiation. Let

$$A = u^3.B + 56 \text{ and } B = (u+1)^2 + 4, \text{ then}$$

$$\begin{cases} c_0 = -11(u^4A + 27u^3B + 28) + 19A; & c_4 = -(2u^4A + 55u^3B) + 84A \\ c_1 = 5(3u^3A + 44u^2B) = 5c_1'; & c_5 = -5(4u^3A + 117u^2B) = -5c_5' \\ c_2 = 25(u^2A + 38uB) = 25c_2'; & c_6 = 25(2u^2A + 41uB) = 25c_6' \\ c_3 = -125(uA + 24B) = -125c_3'; & c_7 = 125.7B = 125c_7' \end{cases}$$

The problem with this representation is that when we recomputed these expressions we discovered that there is a missing term in the expression of c_0. In fact

$$\begin{cases} c_0 = -11u^9 - 22u^8 - 55u^7 - 278u^5 - 1172u^4 - 1390u^3 + 1372 \\ \quad = -11(u^4A + 27u^3B + 28) + 19A + \mathbf{616} \end{cases} \quad (3)$$

We verified also the algorithm presented in Appendix A of [1] where the term f^{616} is missing in the computation of the final exponentiation. Fortunately, the expression of c_0 do not influence the rest of the expressions c_i with $0 < i < 8$. Therefore, we have to add this term to the final result of the hard part of the final exponentiation of the optimal Ate pairing. Using the square-and-multiply algorithm, the additional step f^{616} costs 8 squarings and 3 multiplications in $\mathbb{F}_{p^{16}}$ but we will not add this cost because they are terms precomputed in the algorithm of Zhang et *al.* We will add to their algorithm these operations after the first term of the original algorithm:

$$\begin{cases} A0 \leftarrow T3^8 \\ A1 \leftarrow A0 \cdot T3 \\ A2 \leftarrow A1 \cdot T2 \\ A3 \leftarrow T1^2 \\ A2 \leftarrow A3 \cdot A2 \end{cases} \tag{4}$$

By adding these operations we got in $A2$ the missing term f^{616}. At the end of the algorithm presented by Zhang et *al.* we have to add this term to the final result costing an extra multiplication. So the additional cost is 4 multiplications and 4 squarings in $\mathbb{F}_{p^{16}}$.

Other shortcomings with their algorithm that computed the hard part of the final exponentiation concern the computation of c'_5, c'_0 and c'_4. In fact, in the expression of c'_0, the output of their algorithm is $-11(u^4A+55u^3B+28)+35A$ instead of the result $-11(u^4A+55u^3B+28)+19A$. Also, the expression of c'_4 computed in their algorithm is $-(2u^4A+55u^3B)+148A$ not as mentioned in the development which is $-(2u^4A+55u^3B)+84A$.

The expression of c'_5 is deduced by multiplying the term stocked in the temporary variable T_{11} by the term stocked in F_{14} and not by the one recorded in F_{25}. Also in the computation of c'_7 we must perform the operation $\overline{F_5}.T_4$ instead of $\overline{F_5}.T_6$.

Therefore we must perform some modifications in the original algorithm to have the coherent result at the end. We presented the corrected algorithm in Appendix A, Table 9, and a magma code for the verification of formulas is available in [18]. The additional corrections cost 4 multiplications and 3 squarings in $\mathbb{F}_{p^{16}}$ instead of 3 multiplications and 4 squarings which is the cost of the operations before our modifications. Furthermore Zhang et *al.* claimed that in the final algorithm they used only 16 squarings, but it is not the case because by a simple count we found that one is forced to perform 38 squarings in $\mathbb{F}_{p^{16}}$.

As a consequence to compute the final exponentiation we have to perform 7 exponentiations by u, 2 exponentiations by $(u+1)$, one inversion, 44 cyclotomic squarings in $G_{\phi_2(p^8)}$, 38 multiplications in $\mathbb{F}_{p^{16}}$, 2 cyclotomic cubings in $\mathbb{F}_{p^{16}}$ and p, p^2, p^3, p^4, p^5, p^6, p^6, p^7, p^8-Frobenius maps.

In Table 3 we present the new cost of the final exponentiation of the optimal Ate pairing after our correction of the result of the work in [1]. Hence, by adding some modifications to the original result the overall cost of the optimal Ate pairing on KSS16 curve is 33870M+I. So we have extra 1332 multiplications in $\mathbb{F}_p$ than the cost presented in [1].

Table 3. Complexity of the optimal Ate pairing.

The method	Complexity of Miller loop	Complexity of the final exponentiation
Method of [1]	10208 M	22330M+I
Our correction	10208 M	23662M+I

3 A New Multiple of the Hard Part of the Final Exponentiation

An efficient method to compute the hard part is described by Scott *et al.* [22]. They suggested to write $d = \frac{\phi_k(p)}{r}$ in base p as $d = d_0 + d_1 p + \cdots + d_{\phi(k)-1} p^{\phi(k)-1}$ and find a short vector addition chain to compute f^d much more efficiently than the naive method. In [23], based on the fact that a fixed power of a pairing is still a pairing, Fuentes *et al.* [23] suggested to apply Scott *et al.*'s method with a power of any multiple d' of d with r not dividing d'. This could lead to a more efficient exponentiation than a direct computation of f^d. Their idea of finding the polynomial $d'(x)$ is to apply the LLL-algorithm to the matrix formed by $\mathbb{Q}$-linear combinations of the elements $d(x), xd(x), \ldots, x^{degr-1}d(x)$. In this paper we tried to find a new multiple of $d_1 = 857500 \cdot d$ (with r not dividing d). We use a lattice-based method to find d' such that $f^{d'}$ can be computed in a more efficient way than computing $f^{857500 \cdot d}$.

Thanks to the LLL algorithm [24], the best vector that we found is given by:

$$d'(u) = m_0 + m_1 p + m_2 p^2 + m_3 p^3 + m_4 p^4 + m_5 p^5 + m_6 p^6 + m_7 p^7 = s(u) d_1$$

$$\text{where} \begin{cases} s(u) = u^3/125 \\ m_0 = 2u^8 + 4u^7 + 10u^6 + 55u^4 + 222u^3 + 275u^2 \\ m_1 = -4u^7 - 8u^6 - 20u^5 - 75u^3 - 374u^2 - 375u \\ m_2 = -2u^6 - 4u^5 - 10u^4 - 125u^2 - 362u - 625 \\ m_3 = -u^9 - 2u^8 - 5u^7 - 24u^5 - 104u^4 - 120u^3 + 196 \\ m_4 = u^8 + 2u^7 + 5u^6 + 10u^4 + 76u^3 + 50u^2 \\ m_5 = 3u^7 + 6u^6 + 15u^5 + 100u^3 + 368u^2 + 500u \\ m_6 = -11u^6 - 22u^5 - 55u^4 - 250u^2 - 1116u - 1250 \\ m_7 = 7u^5 + 14u^4 + 35u^3 + 392 \end{cases} \quad (5)$$

Our aim in this section by presenting the new vector d' is to reduce the complexity of computing the hard part of the final exponentiation for the optimal Ate pairing in KSS16 curves and then the complexity of computing the product of n pairings. Let

$$\begin{cases} A = u^3 B + 56 \\ B = (u+1)^2 + 4 \end{cases}$$

then we can write the expressions of m_i where $0 < i < 8$ more simply as follows:

$$\begin{cases} m_0 = 2u^3A + 55u^2B; & m_4 = u^3A + 10u^2B \\ m_1 = -4u^2A - 75uB; & m_5 = 3u^2A + 100uB \\ m_2 = -2uA - 125B; & m_6 = -11uA - 250B \\ m_3 = -u^4A - 24u^3B + 196; & m_7 = 7A \end{cases}$$

These new expressions enabled us to be faster than Zhang et *al.* in the computation of the hard part of the final exponentiation. We detailed the computation of the final exponentiation in the algorithm presented in Appendix A, Table 8, and a magma code for the verification of formulas is available in [18]. The overall cost of this algorithm is then 7 exponentiations by u, 2 exponentiations by $(u+1)$, 34 cyclotomic squarings in $G_{\phi_2(p^8)}$, 32 multiplications in $\mathbb{F}_{p^{16}}$, 3 cyclotomic cubings in $\mathbb{F}_{p^{16}}$ and p, p^2, p^3, p^4, p^5, p^6, p^6, p^7, p^8-Frobenius maps.

Table 4. Comparison between Zhang et *al.* and our new development.

Method	Algorithm	Complexity			
		S_c	M_{16}	F_{16}	C_c
Zhang et al.	1	44	37	8	1
Our development	2	**34**	**32**	**8**	**3**

Our result of computing the hard par of the final exponentiation is compared with the corrected result presented in Sect. 2.3 in Table 4. For a full comparison, we consider the example presented in [1]. The extension tower is built as follows:

- $\mathbb{F}_{p^4} = \mathbb{F}_p[v]/\big(v^4 + 3\big)$
- $\mathbb{F}_{p^8} = \mathbb{F}_{p^4}[w]/\big(w^2 - v\big)$
- $\mathbb{F}_{p^{16}} = \mathbb{F}_{p^8}[z]/\big(z^2 - w\big)$

The cost of operations for computing the optimal Ate pairing on KSS16 curve are presented in Table 4 of [1].

Table 5. Comparison between the two vectors d and d'.

The result	Complexity of algorithm	Complexity of the hard part the final exponentiation
Corrected result of [1]	See cost in Table 8	23537M
Our new algorithm	See cost in Table 9	**22673M**

In Table 5 we compared the complexity in $\mathbb{F}_p$ of our result using a new multiple of the hard part of the final exponentiation and the corrected one of Zhang et *al.* In this table we remark that our computations are faster than those presented in [1] for computing the hard part of the final exponentiation. We saved about 864 multiplications in $\mathbb{F}_p$ which is an interesting result if one is interested in hardware or software implementations of the optimal Ate pairing at the 192-security level.

4 On Computing Products of n Pairings

In some protocols, for example in the BBG HIBE scheme [25], the BLS short group signature scheme [5], ABE scheme due to Waters [14], the non interactive proof systems proposed by Groth and Sahai [26] and others [11,13], it is necessary to compute the product or the quotient of two or more pairings. Scott in [27] and Granger et *al.* in [15] investigated the computation of the product of n pairings.

Let

$$e : G_1 \times G_2 \rightarrow G_3$$

a bilinear non-degenerated map from two additive groups G_1 and G_2 to G_3 a multiplicative group. The evaluation of a product of n pairings is of the form

$$e_n = \prod_{i=1}^{n} e(P_i, Q_i)$$

In this section we are interested by the computation of n pairings. We give a comparison of this computation for different category of curves at the 192-bit security level. For this security level it is recommended by Aranha et *al.* in [9] to use the BLS12 curves to compute the optimal Ate pairing. In this section and in the case where one computes the product of n optimal Ate pairings, we will prove that this category of curves are not a solution for all n specially where $n > 2$. We prove also that the KSS16 curves, proposed as the best solution for computing the product of n pairings by Zhang et *al.* in [1] are not the best for $n = 2$. We First recall in Table 6 the different formulas for the optimal Ate pairing over common families of pairing-friendly curves such as KSS16, KSS18, BN, BLS12 and BLS24 curves. For computing the optimal Ate pairing we have two steps: The Miller loop and the final exponentiation. The computation of the product of n pairings consists only of the computation of the product of n Miller loops followed by the evaluation of the result of the final exponentiation. Recall that in the Miller loop (see the algorithm in Table 2) we have to compute the following step:

$$f \leftarrow f^2 l(Q) \tag{6}$$

Table 6. Optimal Ate pairing on elliptic curves.

Curve	Optimal Ate pairing: $(P, Q) \rightarrow$
KSS16 [1]	$\left((f_{u,Q}(P) l_{[u]Q,[p]Q}(P))^{p^3} l_{Q,Q}(P)\right)^{\frac{p^{16}-1}{r}}$
KSS18 [9]	$\left(f_{u,Q}(P) f_{3,Q}^{p} l_{[u]Q,[3p]Q}(P)\right)^{\frac{p^{18}-1}{r}}$
BN [9]	$\left((f_{6u+2,Q}(P) l_{[6u+2]Q,[p]Q}(P) l_{[6u+2]Q,[-p^2]Q}(P))\right)^{\frac{p^{12}-1}{r}}$
BLS12 [9]	$(f_{u,Q}(P))^{\frac{p^{12}-1}{r}}$
BLS24 [9]	$(f_{u,Q}(P))^{\frac{p^{24}-1}{r}}$

where l is the tangent to the curve at a point depending on Q and depending on the loop iteration in Miller's algorithm. To compute the product of Eq. (6), each doubling function-evaluation step becomes

$$f \leftarrow f^2 \prod_{i=1}^{n} l_i(Q_i) \tag{7}$$

Therefore one needs only to calculate a single squaring in the extension field per doubling rather than n squarings using the naive method of the computation of the product of n pairings.

So to evaluate the cost of the computation of the product of n optimal Ate pairings we have to compute at first:

- **Cost1**: Full squarings in the Miller loop (squarings in Eq. 7).
- **Cost2**: Other operations in the Miller loop (point operations and line evaluation).
- **Cost3**: Final exponentiation.

Then we have to sum **Cost1**, n**Cost2** and **Cost3** to find the overall cost of the product of n pairings.

Table 7. Costs comparison of product of n pairings at the 192-bit security levels.

Costs	KSS16 Zhang	KSS16	BLS12 [10]	BN [9]	KSS18 [9]
Full squarings for DBL	2592M	2592M	5892M	8837M	4158M
Others in Miller loop	7616M	7616M	10760M	16720M	9544M
Final exponentiation	23662M+I	22888M +I	12574M+6I	11145M+6I	23821M+8I
Total cost for $n = 1$	33870M+I	33096M+I	29226M+6I	36702M+6I	37523M+8I
Total cost for $n = 2$	41486M+I	40712M+I	39986M+6I	53422M+6I	47067M+8I
Total cost for $n = 3$	49102M+I	48328M+I	50746M+6I	64567M+6I	56611M+8I
Total cost for $n = 7$	79656M+I	78792M+I	93786M+6I	109147M+6I	94784M +8I

In Table 7, we present the costs for computing the product of n pairings considering common curves in Table 6. From Table 7, we can deduce that for $n = 2$, meaning when we would like to compute the product of two parings, it is better to use BLS12 curves. In the case of $n > 2$ as mentioned in [1] KSS16 curves can give the fastest computations of products or quotients of n pairings.

Security of Cryptographic protocols is important in practice. That's why, when we compute optimal Ate pairing on KSS16 curves we have to verify the security of the parameters of the elliptic curve. In the next section we will present a detailed study of the security of the computation of the optimal Ate pairing and more precisely the resistance against the subgroup attacks.

5 Subgroup Security for KSS16 Pairing-Friendly Curves

A detailed study on subgroup security for pairing-friendly curves was recently studied by Baretto et *al.* [17]. They focus on common families of elliptic curves having twists of order six such as BN, BL12, BLS24 and KSS18 curves. In particular they provided parameters that enable the aforementioned curves to be resistant against subgroups attacks. In this section, we extend the same analysis to the KSS family of elliptic curves having quartic twists and of embedding degree 16. We first recall the definition of *subgroup secure curves* concept from [17] The subgroup security concept explicitly described on pairing-friendly curves by Barreto et *al.* [17], is a property that strengthens the resistance of pairing-friendly curves against subgroup attacks. Let E be an elliptic curve of embedding degree k and parameterised by $p(u)$, $t(u)$, $r(u) \in \mathbb{Q}[u]$. Let d be the degree of the twist of the elliptic curve E and let $E'(\mathbb{F}_{p^{k/d}})$ its twists. Let $h_1(u) = \dfrac{\mid E(\mathbb{F}_p)(u) \mid}{r(u)}$, $h_2(u) = \dfrac{\mid E'(\mathbb{F}_{p^{k/d}})(u) \mid}{r(u)}$ and $h_T = \dfrac{\mid G_{\phi_k}(p(u)) \mid}{r(u)}$ be the indices of the three groups on which a pairing is defined.

Definition 3. [17] *The curve E is* subgroup secure *if all $\mathbb{Q}[u]$-irreducible factors of $h_1(u)$, $h_2(u)$, $h_T(u)$ that represent primes and that have degree at least the degree of $r(u)$, contain no prime factor smaller than $r(u_0) \in \mathbb{Z}$ when evaluated at $u = u_0$.*

In the case of KSS16, the indices are given in the following proposition:

Proposition 4. *Let $p(u)$, $t(u)$, $r(u) \in \mathbb{Q}[u]$ be the parameters of the KSS16 pairing-friendly elliptic curve. The indice $h_T = \dfrac{p(u)^8+1}{r(u)}$ is a polynomial in u of degree 72. Also $h_1(u) = (125/2)(u^2+2u+5)$ and the order of the quartic twist $E'(\mathbb{F}_{p^4})$ is $\mid E'(\mathbb{F}_{p^4}) \mid = h_2(u) \cdot r(u)$ where $h_2(u) = (1/15059072)(u^{32} + 8u^{31} + 44u^{30} + 152u^29 + 550u^{28} + 2136u^{27} + 8780u^{26} + 28936u^{25} + 83108u^{24} + 236072u^{23} + 754020u^{22} + 2287480u^{21} + 5986066u^{20} + 14139064u^{19} + 35932740u^{18} + 97017000u^{17} + 237924870u^{16} + 498534968u^{15} + 1023955620u^{14} + 2353482920u^{13} + 5383092978u^{12} + 10357467880u^{11} + 17391227652u^{10} + 31819075896u^{9} + 65442538660u^{8} + 117077934360u^{7} + 162104974700u^{6} + 208762740168u^{5} + 338870825094u^{4} + 552745197960u^{3} + 632358687500u^{2} + 414961135000u + 126854087873)$.*

Proof. The order of the group $E(\mathbb{F}_{p^4})$ is $\mid E(\mathbb{F}_{p^4}) \mid = p^4 + 1 - t_4$ where $t_4 = t^4 - 4pt^2 + 2p^2$ (see [28, Theorem 4.12]). The order of the correct quartic twist $E'(\mathbb{F}_{p^4})$ is given by $\mid E'(\mathbb{F}_{p^4}) \mid = p^4 + 1 + v_4$ where $v_4^2 = 4p^4 - t_4^2$ (see [29, Proposition 2]). A direct calculation gives the cofactor as $h_2(u) = \frac{p^4+1+v_4}{r(u)}$.

Remark 5. *The value used in [1] for the computation of optimal pairing on* $KSS16$ *curves is* $u_0 = 2^{49}+2^{26}+2^{15}-2^7-1$. *With this value we see that* $h_2(u_0)$ *has the factorisation* $2\cdot1249\cdot366593\cdot c_{1515}$ *where* c_{1515} *is still a composite integer of* 1515 *bits. This means that the corresponding curve fails to satisfy the small subgroup attack property. In the following section we search for a parameter* u *to avoid subgroup attack on this curve.*

For the 192-bit security level, the u_0 which gives corresponding sizes of r and p must be an integer of bit size at least 49. Also, the good u_0 must be such that $p(u_0), r(u_0)$, $h_2(u_0)$ and $h_T(u_0)$ are simultaneously prime. Since $u \equiv \pm 25 \bmod 70$ (for p to represent integers) one can easily see that $h_2(u) \equiv 0 \bmod 2$ and $h_T(u) \equiv 0 \bmod 2$. We will therefore search for u_0 such that $p(u_0), r(u_0)$, $h_2(u_0)/2$ and $h_T(u_0)/2$ are simultaneously prime. One can have a chance to obtain such a u_0 if and only if those polynomials satisfy the Bunyakovsky's property. A quick verification enables to see that the prime number 17 divides these polynomials when evaluated at $n \in \mathbb{N}$. Therefore it is enough to search for prime numbers with 2 and/or 17 as factors. The Batemann-Horn conjecture then ensures that they are approximately 24500 values of $u_0 \in [2^{49}, 2^{53}]$ with $p(u_0), r'(u_0), h'_2(u_0)$ and $h'_T(u_0)$ simultaneously prime, where $r(u) = 17^{n_1}\cdot r'(u)$, $h_2(u) = 2\cdot 17^{n_2}\cdot h'_2(u)$ and $h_T = 2\cdot 17^{n_3}\cdot h'_T(u)$ for some positive or zero integers n_1, n_2 and n_3. A careful search enabled us, after several long tries starting with x_0 of Hamming weight 5, to obtain the following value

$$u_0 = 2^{50} + 2^{47} - 2^{38} + 2^{32} + 2^{25} - 2^{15} - 2^5 - 1$$

which gives a prime p of 492 bits, $r(u_0) = r'(u_0)$ prime of 386 bits, $h_2(u_0) = 2\cdot17\cdot h'_2(u_0)$ and $h_T = 2\cdot17\cdot h'_T(u_0)$ where $h'_2(u_0)$ and $h'_T(u_0)$ are prime numbers of 3544 bits and 1577 bits respectively. For the value of p obtained the extension field $\mathbb{F}_{p^{16}}$ is built using the following tower of extensions:

- $\mathbb{F}_{p^2} = \mathbb{F}_p[\alpha]/(\alpha^2 - 11)$
- $\mathbb{F}_{p^4} = \mathbb{F}_{p^2}[\beta]/\big(\beta^2 - \alpha)\big)$
- $\mathbb{F}_{p^8} = \mathbb{F}_{p^4}[\gamma]/\big(\gamma^2 - \beta\big)$
- $\mathbb{F}_{p^{16}} = \mathbb{F}_{p^8}[\theta]/\big(\theta^2 - \gamma\big)$

An example of elliptic curve E over $\mathbb{F}_p$ that satisfies $|E(\mathbb{F}_p)| = p+1-t$ has the equation $E:\ y^2 = x^3 + 17x$. The corresponding quartic twist E' over $\mathbb{F}_{p^4}$ with order $|E'(F_{p^4})| = 2\cdot17\cdot h'_2(u_0)\cdot r(u_0)$ is the curve $E':\ y^2 = x^3 + 17/\beta x$.

6 Conclusion

In many pairing-based protocols the evaluation of the product or the quotient of many pairings is required. In this paper we were interested in the computation of the product of n optimal Ate pairings at the high security level.

This problem was first considered by Zhang et *al.* [1]. They suggested the KSS16 curves as a best choice for computing n pairings. We checked their results on the computation of the hard part of the final exponentiation of the optimal Ate pairing. We found that they missed 1332 multiplications in $\mathbb{F}_p$ in their complexity calculation. We corrected their algorithm and we presented a new algorithm for the computation of the final exponentiation based on a new multiple of the hard part of the final exponentiation. With this new vector we saved about 864 multiplications in the basic field which is an important result if one thinks about hardware or software implementations. We implemented our new algorithms in Magma to verify their correctness [18]. We computed also the product of n pairings. We proved that for $n = 2$ it is better to use BLS12 curves and for $n > 2$ KSS16 curves are the best solution. Finally we proposed a new parameter u for the KSS16 curves to ensure the resistance against the small subgroup attacks.

A Algorithms

In these tables and to have the same expressions as Zhang et *al.* we denote by f the result of Miller loop and by M the result of the first part of the final exponentiation.

Table 8. Final exponentiation with a new exponent. See [18] for the magma code for the verification.

Operations	Terms computed	Cost
$E1 = f^{p^8} E2 = E1 \cdot f^{-1}$	$M = f^{p^8-1}$	
$T0 = M^2; T1 = T0^2$	$M^2; M^4$	$2S_{16}$
$T2 = M^{u+1}; T3 = T2^{u+1}$	$M^{u+1}; M^{(u+1)^2}$	$2E_u$
$T4 = T3 \cdot T1$	$M^{(u+1)^2+4} = M^B$	$1M_{16}$
$T5 = T4^u; T6 = T4^5$	$M^{uB}; M^{5B}$	$1E_u + 1M_{16} + 2S_{16}$
$T7 = T1^8; T8 = T7^2$	$M^{32}; M^{64}$	$4S_{16}$
$T9 = T7 \cdot T1^{-1}; T10 = T9^2$	$M^{28}; M^{56}$	$1M_{16} + 1S_{16}$
$T11 = T5^u; T12 = T11^u$	$M^{u^2B}; M^{u^3B}$	$2E_u$
$T01 = T12 \cdot T10$	$M^{u^3B+56} = M^A$	$1M_{16}$
$T14 = T01^u; T13 = T14^{-2}$	$M^{uA}; M^{-2uA}$	$1E_u + 1S_{16}$
$T00 = T6^5; T15 = T00^5$	$M^{25B}; M^{125B}$	$2M_{16} + 4S_{16}$
$T0 = T13 \cdot T15^{-1}$	$M^{-2uA-125B} = M^{c_2}$	$1M_{16}$
$T16 = T0^2; T17 = T13^4$	$M^{2c_2}; M^{-8uA}$	$3S_{16}$
$T18 = T17 \cdot T14$	M^{-7uA}	$1M_{16}$
$T2 = T16 \cdot T18$	$M^{2c_2-7uA} = M^{c_6}$	$1M_{16}$
$T19 = T14^u; T20 = T19^u$	$M^{u^2A}; M^{u^3A}$	$2E_u$
$T21 = T20^u; T22 = T19^2$	$M^{u^4}; M^{2u^2A}$	$1E_u + 1S_{16}$
$T23 = T5^5; T24 = T23^5$	$M^{5uB}; M^{25uB}$	$2M_{16} + 4S_{16}$

Table 8. (*continued*)

Operations	Terms computed	Cost
$T25 = T24^3; T26 = T24 \cdot T25$	$M^{75uB}; M^{100uB}$	$1C_{16} + 1M_{16}$
$T27 = T22^2$	M^{4u^2A}	$1S_{16}$
$T37 = (T27 \cdot T25)^{-1}$	$M^{-4u^2A-75uB} = M^{c_1}$	$1M_{16}$
$T28 = T27 \cdot T19^{-1}$	M^{3u^2A}	$1M_{16}$
$T3 = T28 \cdot T26$	$M^{3u^2A+100xB} = M^{c_5}$	$1M_{16}$
$T29 = T11^5; T30 = T29^2$	$M^{5u^2B}; M^{10u^2B}$	$1M_{16} + 3S_{16}$
$T4 = T20 \cdot T30$	$M^{u^3A+10u^2B} = M^{c_4}$	$1M_{16}$
$S0 = T20^2; S1 = T30^5$	$M^{2u^3A}; M^{50u^2B}$	$1M_{16} + 3S_{16}$
$S2 = S1 \cdot T29; S3 = S0 \cdot S2$	$M^{55u^2B}; M^{2u^3A-55u^2B} = M^{c_0}$	$2M_{16}$
$T31 = T12^{24}$	M^{24u^3B}	$1C_{16} + 3S_{16}$
$T5 = T21^{-1} \cdot T31^{-1}$	$M^{-u^4A-24u^3B}$	$1M_{16}$
$T6 = T8^3 \cdot T1$	M^{196}	$1M_{16} + 1C_{16}$
$T7 = T5 \cdot T6$	$M^{-u^4A-24u^3B+196} = M^{c_3}$	$1M_{16}$
$T8 = T1^7$	$M^{7A} = M^{c_7}$	$2M_{16} + 2S_{16}$
$T32 = T37^p \cdot T7^{p^3} \cdot T3^{p^5} \cdot T8^{p^7}$	$M^{c_1p+c_3p^3+c_5p^5+c_7p^7}$	$3M_{16} + 4(15M)$
$T33 = T0^{p^2} \cdot T2^{p^6}$	$M^{c_2p^2+c_6p^6}$	$1M_{16} + 2(12M)$
$T = S3 \cdot T32 \cdot T33 \cdot T4^{p^4}$	$M^{\frac{p^8+1}{r}}$	$3M_{16} + 1(8M)$

Table 9. Corrected version of the final exponentiation in [1]. See [18] for the magma code for the verification.

Operations	Terms computed	Cost
$E1 = f^{p^8} E2 = E1 \cdot f^{-1}$	$M = f^{p^8-1}$	
$T1 = E2^4; T2 = T1^8; T3 = T2^2$		$6S_{16}$
$A0 = T3^8; A1 = A0 \cdot T3$		$1M_{16} + 3S_{16}$
$A2 = A1 \cdot T2; A3 = T1^2$		$1M_{16} + 1S_{16}$
$A2 = A3 \cdot A2$		$1M_{16}$
$F1 = T2 \cdot T1^{-1}; F2 = F1^2$		$1M_{16} + 1S_{16}$
$F3 = E2^{u+1}; F4 = F3^{u+1}$		$2E_{u+1}$
$F5 = F4 \cdot T1; T4 = F5^8$	$F5 = M^B$	$1M_{16} + 3S_{16}$
$F6 = F5^u; F7 = F5^{-1} \cdot T4$	$F7 = M^{c_7'}$	$1E_u + 1M_{16}$
$F8 = T4^3; T5 = F6^8$		$1C_{16} + 3S_{16}$
$F9 = F6^u; F10 = T5 \cdot F6^{-1}$		$1E_u + 1M_{16}$
$F11 = F10^2; T6 = F9^8$		$4S_{16}$
$F12 = F9^u; F13 = T6 \cdot F9^{-1}$		$1E_u + 1M_{16}$
$F14 = F13^2; F15 = F12 \cdot F2$	$F15 = M^A$	$1S_{16} + 1M_{16}$

Table 9. (*continued*)

Operations	Terms computed	Cost
$T7 = F15^2; T8 = T7^4$		$3S_{16}$
$S1 = T8^2; S2 = T7^2$		$2S_{16}$
$S3 = S2 \cdot S1; S4 = S3 \cdot F15^{-1}$		$2M_{16}$
$T9 = S1^4; S5 = S3 \cdot T9$		$1M_{16} + 2S_{16}$
$S6 = F14^2; F16 = F15^u$		$1E_u + 1S_{16}$
$F22 = F16 \cdot F8$	$F22 = M^{c_3'}$	$1M_{16}$
$F23 = F22^u; F24 = F23 \cdot F11$	$F24 = M^{c_2'}$	$1E_u + 1M_{16}$
$T10 = F23^2; F25 = F23^u$		$1E_u + 1S_{16}$
$F26 = T10 \cdot F10^{-1}; T11 = F25^4$	$F26 = M^{c_6'}$	$1M_{16} + 2S_{16}$
$F27 = F25^u; F28 = T11 \cdot F25^{-1}$		$1E_u + 1M_{16}$
$F29 = F13 \cdot F14; F30 = T11 \cdot F29$	$F30 = M^{c_5'}$	$2M_{16}$
$F31 = F28 \cdot S6^{-1}; F32 = F12^2$		$1M_{16} + 1S_{16}$
$F33 = F32 \cdot F12; F34 = F27 \cdot F33$		$2M_{16}$
$F35 = F34^2; F36 = F35 \cdot F12$		$1M_{16} + 1S_{16}$
$F37 = F36^{-1} \cdot S5; F38 = F34 \cdot F1$	$F37 = M^{c_4'}$	$2M_{16}$
$F39 = F38^2; F40 = F39^2$		$2S_{16}$
$F41 = F40^2; F42 = F39 \cdot F38$		$1M_{16} + 1S_{16}$
$F43 = F41 \cdot F42; F44 = F43^{-1} \cdot S4$		$2M_{16}$
$H1 = F7^{p^7}; H2 = F22^{p^3}$		$2(14M)$
$H3 = F24^{p^2}; H4 = F26^{p^6}$		$2(12M)$
$H5 = F30^{p^5}; H6 = F31^p$		$2(14M)$
$H7 = F37^{p^4}; H8 = H1 \cdot H2^{-1}$		$1M_{16} + 1(8M)$
$H9 = H8^2; H10 = H9^2$		$2S_{16}$
$H11 = H10 \cdot H8; H12 = H11 \cdot H3$		$2M_{16}$
$H13 = H12 \cdot H4; H14 = H13^2$		$1M_{16} + 1S_{16}$
$H15 = H14^2; H16 = H15 \cdot H13$		$1M_{16} + 1S_{16}$
$H17 = H16 \cdot H6; H18 = H17 \cdot H5^{-1}$		$2M_{16}$
$H19 = H18^2; H20 = H19^2$		$2S_{16}$
$H21 = H20 \cdot H18; H22 = H21 \cdot H7$		$2M_{16}$
$H23 = H22 \cdot F44$	$H23 = M^{d'}$	$1M_{16}$

References

1. Zhang, X., Lin, D.: Analysis of optimum pairing products at high security levels. In: Galbraith, S., Nandi, M. (eds.) INDOCRYPT 2012. LNCS, vol. 7668, pp. 412–430. Springer, Heidelberg (2012). doi:10.1007/978-3-642-34931-7_24
2. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). doi:10.1007/3-540-44647-8_13

3. Cocks, C.: An identity based encryption scheme based on quadratic residues. In: Honary, B. (ed.) Cryptography and Coding 2001. LNCS, vol. 2260, pp. 360–363. Springer, Heidelberg (2001). doi:10.1007/3-540-45325-3_32
4. Libert, B., Quisquater, J.-J.: Identity based undeniable signatures. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 112–125. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24660-2_9
5. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the Weil pairing. J. Cryptol. **17**(4), 297–319 (2004)
6. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Linawati, Mahendra, M.S., Neuhold, E.J., Tjoa, A.M., You, I. (eds.) ICT-EurAsia 2014. LNCS, vol. 8407, pp. 89–98. Springer, Heidelberg (2006). doi:10.1007/978-3-642-55032-4_60
7. Barreto, P.S.L.M., Naehrig, M.: Pairing-friendly elliptic curves of prime order. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 319–331. Springer, Heidelberg (2006). doi:10.1007/11693383_22
8. Barreto, P.S.L.M., Lynn, B., Scott, M.: Constructing elliptic curves with prescribed embedding degrees. In: Cimato, S., Persiano, G., Galdi, C. (eds.) SCN 2002. LNCS, vol. 2576, pp. 257–267. Springer, Heidelberg (2003). doi:10.1007/3-540-36413-7_19
9. Aranha, D.F., Fuentes-Castañeda, L., Knapp, E., Menezes, A., Rodríguez-Henríquez, F.: Implementing pairings at the 192-bit security level. In: Abdalla, M., Lange, T. (eds.) Pairing 2012. LNCS, vol. 7708, pp. 177–195. Springer, Heidelberg (2013). doi:10.1007/978-3-642-36334-4_11
10. Ghammam, L., Fouotsa, E.: On the computation of the optimal ate pairing at the 192-bit security level. IACR Cryptology ePrint Archive, 2016:130 (2016)
11. Chen, L., Cheng, Z., Smart, N.P.: A built-in decisional function and security proof of id-based key agreement protocols from pairings. IACR Cryptology ePrint Archive, 2006:160 (2006)
12. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004). doi:10.1007/978-3-540-28628-8_3
13. Abdalla, M., Catalano, D., Dent, A.W., Malone-Lee, J., Neven, G., Smart, N.P.: Identity-based encryption gone wild. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 300–311. Springer, Heidelberg (2006). doi:10.1007/11787006_26
14. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005). doi:10.1007/11426639_7
15. Granger, R., Smart, N.P.: On computing products of pairings. IACR Cryptology ePrint Archive, 2006:172 (2006)
16. Kachisa, E.J., Schaefer, E.F., Scott, M.: Constructing Brezing-Weng pairing friendly elliptic curves using elements in the cyclotomic field. IACR Cryptology ePrint Archive 2007:452 (2007)
17. Barreto, P.S.L.M., Costello, C., Misoczki, R., Naehrig, M., Pereira, G.C.C.F., Zanon, G.: Subgroup security in pairing-based cryptography. In: Lauter, K., Rodríguez-Henríquez, F. (eds.) LATINCRYPT 2015. LNCS, vol. 9230, pp. 245–265. Springer, Cham (2015). doi:10.1007/978-3-319-22174-8_14
18. Fouotsa, E., Ghammam, L.: http://www.camercrypt.org/KSS16-finalexponentiation
19. Kim, T.: Extended tower number field sieve: a new complexity for medium prime case. IACR Cryptology ePrint Archive, 2015:1027 (2015)

20. Jeong, J., Kim, T.: Extended tower number field sieve with application to finite fields of arbitrary composite extension degree. IACR Cryptology ePrint Archive, 2016:526 (2016)
21. Miller, V.S.: The Weil pairing, and its efficient calculation. J. Cryptol. **17**(4), 235–261 (2004)
22. Scott, M., Benger, N., Charlemagne, M., Dominguez Perez, L.J., Kachisa, E.J.: On the final exponentiation for calculating pairings on ordinary elliptic curves. In: Shacham, H., Waters, B. (eds.) Pairing 2009. LNCS, vol. 5671, pp. 78–88. Springer, Heidelberg (2009). doi:10.1007/978-3-642-03298-1_6
23. Fuentes-Castañeda, L., Knapp, E., Rodríguez-Henríquez, F.: Faster hashing to $\mathbb{G}_2$. In: Miri, A., Vaudenay, S. (eds.) SAC 2011. LNCS, vol. 7118, pp. 412–430. Springer, Heidelberg (2012). doi:10.1007/978-3-642-28496-0_25
24. Smeets, I., Lenstra, A.K., Lenstra, H., Lovász, L., van Emde Boas, P.: The history of the LLL-algorithm. In: Nguyen, P.Q., Vallée, B. (eds.) The LLL Algorithm - Survey and Applications, pp. 1–17. Springer, Heidelberg (2010)
25. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005). doi:10.1007/11426639_26
26. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008). doi:10.1007/978-3-540-78967-3_24
27. Scott, M.: Computing the tate pairing. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 293–304. Springer, Heidelberg (2005). doi:10.1007/978-3-540-30574-3_20
28. Washington, L.C.: Elliptic Curves Number Theory and Cryptography. Discrete Mathematics and Its Applications. Chapman and Hall, London (2008)
29. Hesse, F., Smart, N.P., Vercauteren, F.: The eta pairing revisited. IEEE Trans. Inf. Theory **52**(10), 4595–4602 (2006)